VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zabezpečený obousměrný rádiový kanál s vysílacími moduly

Secured Bidirectional Radio Channel with Transceiver Modules

2016

Bc. Adam Deštěnský

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání diplomové práce

Student: **Bc. Adam Deštěnský**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2601T013 Telekomunikační technika

Téma: Zabezpečený obousměrný rádiový kanál s vysílacími moduly
Secured Bidirectional Radio Channel with Transceiver Modules

Jazyk vypracování: čeština

Zásady pro vypracování:

Student má za úkol vytvořit pomocí integrovaných bezdrátových modulů obousměrný rádiový kanál mezi alespoň dvěma jednočipy. Rádiový kanál bude programově zabezpečen proti chybám při přenosu a šifrován obdobou některého ze známých standardů.

Vypracovaná práce bude splňovat následující body zadání:
1. Popis funkce komunikačních bezdrátových modulů a jejich ovládání.
2. Sestavení oddělených testovacích zapojení jednočipů s komunikačními moduly a oživení obousměrné komunikace mezi moduly.
3. Zabezpečení přenášených dat proti chybám při přenosu s případným opakováním přenosu.
4. Šifrování přenášených dat.
5. Testování.

Seznam doporučené odborné literatury:

MATOUŠEK, D. *Práce s mikrokontroléry ATMEL AVR.* 1. vyd. Praha: BEN - technická literatura, c2006, 319 s. ISBN 80-730-0174-8.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Martin Tomis**

Datum zadání: 01.09.2013
Datum odevzdání: 29.04.2016

doc. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou/diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: *25. dubna 2016*

.....

podpis studenta

Poděkování

Rád bych poděkoval Ing. Martinu Tomisovi za odbornou pomoc a konzultaci při vytváření této diplomové práce a přítelkyni Vlastě Mlčochové za vytrvalou podporu a trpělivost.

Abstrakt

Práce shrnuje možnosti zabezpečení rádiového kanálu proti chybám a odposlechu pomocí známých algoritmů a nastiňuje problematiku spojení v ISM pásmu. V další části je navržen systém řízení datového spojení včetně odpovídající struktury rádiového paketu. V následující kapitole se zabývá konstrukcí mikropočítačem řízených radiostanic v pásmu 433 MHz pro přenos dat. Podrobně je rozebrán návrh hardwaru i softwaru s cílem zkonstruovat rádiová koncová zařízení ovládaná z osobního počítače pomocí sériové linky RS232. Závěr práce obsahuje výsledky provedených testů a měření přenosu dat v reálném prostředí včetně vyhodnocení.

Klíčová slova

ARQ, FEC, Hammingův kód, CRC-8, ISM, AES, Atmel AVR, RFM12BP

Abstract

This thesis summarizes methods of coding radio channel against errors and eavesdropping by known algorithms and outlines problems in ISM band. The following part of thesis contains proposal of the communication protocol, including specifications of radio packet. Second chapter is dedicated to the development of a radio station for data transfer in 433 MHz band, controlled by single-chip microcontroller. Proposals of hardware and software design are described in details with the goal to manufacture end device managed by RS232. The last part of thesis contains a results of measurement in the real conditions.

Key words

ARQ, FEC, Hamming code, CRC-8, ISM, AES, Atmel AVR, RFM12BP

Obsah

Seznam použitých symbolů

Symbol	Jednotky	Význam symbolu
B	Hz	Šířka pásma
C	bit/s	Informační kapacita
CR	-	Kompresní poměr
N	W	Výkon šumu
n	bit	Počet bitů
R	-	Rychlost kódování
S	W	Střední hodnota výkonu užitečného signálu
V	bit/s	Přenosová rychlost

Seznam použitých zkratek

Zkratka	Význam
A/D	Analog to Digital
AES	Advanced Encryption Standard
AFC	Automatic Frequency Control
ARQ	Automatic Request Query
ASCII	American Standard Code for Information Interchange
BER	Bit Error Rate
CLK	Clock
CPU	Central Processing Unit
CRC	Cyclic Redundancy Code
ČTÚ	Český telekomunikační úřad
DES	Data Encryption Standard
DIL	Dual In Line
DPS	Deska plošných spojů
EEPROM	Electrically Erasable Programmable Read Only Memory
e.r.p	Effective Radiated Power
ESR	Equivalent Series Resistance
FEC	Forward Error Correction
FIFO	First In Fisrt Out
FSK	Frequency Shift Keying
GND	Ground
IC	Integrated Circuit
IDE	Integrated Development Environment
I/O	Input/Output
ITU	International Telecommunication Union
ISM	Industrial Scientific and Medical
LNA	Low Noise Amplifier
LPC	Linear Predictive Coding
LSB	Least Significant Bit

MCU	Microcontroller Unit
MISO	Master Input Slave Output
MOSI	Master Output Slave Input
MSB	Most Significant Bit
PCM	Pulse Code Modulation
PLL	Phase Locked Loop
POR	Power On Reset
RAM	Random Access Memory
RPE	Regular Pulse Excitation
RXEN	Receiver Enable
SB	Servisní bajt
SDI	Serial Data Input
SDO	Serial Data Output
SNR	Signal to Noise Ratio
SPI	Serial Peripheral Interface
SS	Slave Select
TCP/IP	Transmission Control Protocol/Internet Protocol
THT	Trough Hole Technology
TTL	Transistor - Transistor Logic
TXEN	Transmitter Enable
USART	Universal Synchronous and Asynchronous Serial Receiver and Transmitter
VDI	Valid Data Indicator
VF	Vysokofrekvenční
WCDMA	Wideband Code Division Multiple Access
Wi-Fi	Wireless Fidelity
WPA	Wi-Fi Protected Access
XOR	Exclusive OR

Seznam ilustrací a seznam tabulek

Číslo tabulky	Název tabulky	Číslo stránky
1.1	Přehled pásma ISM	22
1.2	Generující polynomy pro CRC-8	24
1.3	Look-up tabulka Hammingova kódu (8,4)	26
1.4	Srovnání šifrovacích algoritmů	28
1.5	Kombinace režimů spojení	31
1.6	Hodnoty servisního bajtu	32
2.1	Rádiové a elektrické parametry modulu RFM12BP-433	37
2.2	Instrukce pro obsluhu RFM12BP	40
2.3	Význam jednotlivých bitů STATUS registru	41
2.4	Naměřené hodnoty dosahu rádiového spojení	53
2.5	Doba přenosu 20 KiB souboru	55
2.6	Ukázka šifrování textu	58

Úvod

V současné době jsou malé digitální komunikační moduly využívány ve stále větším množství elektroniky, a to jak v domácnosti, tak v průmyslu. Jejich výhodné vlastnosti a nízká cena, je předurčují k nasazení v oblastech, jako je například sběr telemetrických dat, automatizace, dálkové řízení a mnoha dalších aplikací, kde požadujeme bezdrátový přenos menšího objemu dat. Především u spotřební elektroniky je trend zcela zřejmý a již dnes je v mnoha domácnostech používána typicky meteostanice s vnější radiovou měřící jednotkou, bezdrátový zvonek a pokojový termostat. Vzhledem k takto širokému poli působnosti ale dochází ke vzájemnému působení jednotlivých komunikačních kanálů mezi sebou. I přesto, že výrobci na tuto možnost myslí, a svá zařízení vybavují různými mechanismy vzájemné synchronizace, dochází občas ke vzájemnému nežádoucímu rušení.

V teoretickém rozboru se zabýváme nejprve teorií přenosu zpráv, kde jsou objasněny základní pojmy a naznačen problém působení vnějších vlivů na přenosový kanál. Dále se věnujeme rádiovému komunikačnímu řetězci tak, jak jej definoval Claude Shannon. Zde je již naznačeno, jaké kroky jsou potřebné pro provedení zabezpečeného radiového přenosu. Následuje popis kmitočtového pásma, pro průmyslové, vědecké a lékařské využití, jeho hlavní výhody a nevýhody. V další části jsou popsány konkrétní metody zabezpečení radiového kanálu proti chybám, způsobeným vnějšími vlivy (korekce chyb) a proti záměrnému odposlechu (šifrování). Závěrečná část rozboru naznačuje problematiku řízení rádiového přenosu s volbou struktury odesílaných dat.

Praktická část práce je věnována konstrukci radiové stanice s využitím rádiového modulu RFM12BP-433 od čínské firmy Hope Microelectronics. Nejprve jsou podrobně popsány jednotlivé funkční bloky radiostanice s volbou komponentů. Následuje podrobný popis vlastností rádiového modulu, jeho radiová část a rozhraní pro komunikaci s mikrokontrolérem. Navazujícím tématem je dále samotná konstrukce radiostanice, a to jak její hardwarové části, tak softwaru pro řídící mikrokontrolér.

Závěrečná část si klade za cíl provést důkladné testování radiového přenosu, jak nezabezpečeného, tak ve všech implementovaných režimech zabezpečení. Během testů bude dále zkoumán vliv dalších nastavitelných parametrů radiostanice na kvalitu přenosu, jako je vysílací výkon, přenosová rychlost a citlivost přijímače. Nedílnou součástí testů bude také test maximálního dosahu radiostanic, a to jak pro výkon stanovený nařízením ČTÚ, tak pro maximální nastavitelný výkon.

1 Teoretický rozbor

1.1 Teorie přenosu informace

Hlavním úkolem sdělovací techniky je umožnit lidem vzájemný přenos informací (zpráv) na takové vzdálenosti, které již nedokážou překlenout vrozenými smysly (řeč, sluch, zrak). Dalším úkolem je vytvořit příznivé podmínky, za kterých mohou komunikovat lidé se stroji, případně stroje vzájemně mezi sebou (automatizace).

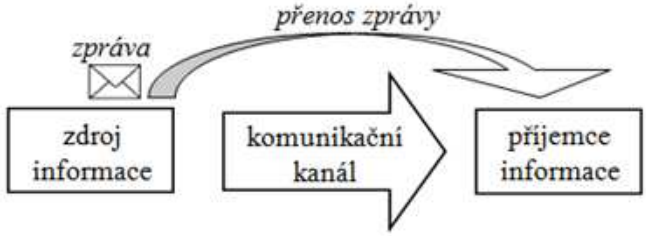
1.1.1 Základní pojmy

Definice základních pojmů z oblasti zpracování signálů jsou poměrně rozsáhlé, my si vystačíme s definicemi uvedenými v [1]:

- **Informace**: v nejširším slova smyslu je chápána jako obraz reálného světa, o jeho stavu a procesech, které v něm probíhají.
- **Zpráva**: oboustranně neomezená posloupnost symbolů nějaké abstraktní abecedy, vysílaná zdrojem informace a přenášena komunikačním kanálem. Představuje sestavu prvků nebo znaků, které nesou významově ucelenou informaci.
- **Abeceda**: prvky (elementy), z nichž je zpráva sestavena. Může být nespojitá (například binární kód) nebo spojitá (například fyzikální veličina teploty).
- **Signál**: zpráva převedena do konkrétní fyzikální formy, která je vhodná pro přenos určitým prostředím.
- **Systém přenosu informace**: komunikační řetězec, který se skládá ze tří základních systémových prvků a to: zdroje informace, komunikačního kanálu a spotřebiče informace.

1.1.2 Princip přenosu zpráv

Obecné schéma komunikace mezi zdrojem a příjemcem informace je uvedeno na následujícím obrázku.



Obrázek 1.1: *Blokové schéma systému přenosu zpráv (převzato z [1])*

Toto schéma představuje nejjednodušší komunikační řetězec. Zdroj musí informaci pomocí definované abecedy převést do podoby zprávy tak, aby byla příjemcem čitelná. V dalším kroku zprávu převede na soustavu signálů, vhodných pro přenos dostupným komunikačním kanálem. Ten může být tvořen drátovým nebo bezdrátovým médiem.

Neméně důležité je v případě bezpečnosti i opatření proti odposlechu zprávy a následnému dekódování informace třetí osobou v komunikačním kanále.

1.1.3 Komunikační kanál

Kanál vytváří přenosové prostředí mezi zdrojem a příjemcem zprávy. Základní charakteristikou kanálu je, jaké množství informace za jednotku času dokáže přenést. Pro tuto vlastnost používáme název informační kapacita nebo také propustnost kanálu [1]. Musí platit:

$$v_{pr} \leq C \tag{1.1}$$

kde *v* je přenosová rychlost v bit/s a *C* informační kapacita v bit/s.

Kanály můžeme rozdělit do dvou kategorií:

- analogový (spojitý) kanál,
- digitální kanál.

1.1.3.1 ***Analogový (spojitý) kanál***

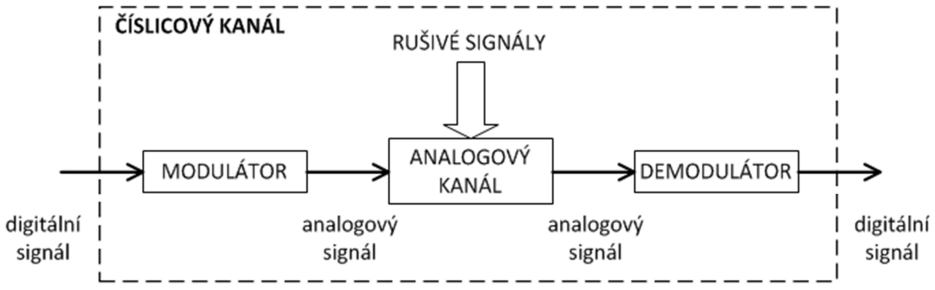
Datová komunikace se velmi často realizuje na analogových kanálech, ať už z historických důvodů (stávající telefonní sítě) nebo z fyzikálních (radiový přenos). U těchto kanálů musíme rovněž znát jejich informační kapacitu. K tomu nám poslouží vztah, vycházející z Shannon – Hartleyova zákona, který určuje nejvyšší dosažitelnou přenosovou kapacitu [2]. Platí:

$$\mathrm{C} = B.log_2.\left(1 + \frac{S}{N}\right) \tag{1.2}$$

kde *C* je informační kapacita, *B* šířka pásma v Hz, *S* výkon užitečného signálu ve W a *N* výkon šumu ve W.

1.1.3.2 ***Digitální komunikační kanál***

Při přenosu digitálního signálu vzniká problém, jak takovýto signál odeslat příjemci, pokud máme k dispozici analogový komunikační kanál. Ten je pro neschopnost přenést stejnosměrnou složku nevhodný pro digitální signály. Řešení nabízí modulátor. Schéma komunikačního řetězce s modulátorem ukazuje blokové schéma.



Obrázek 1.2: *Přenos digitálního signálu analogovým kanálem (převzato z [1])*

Modulace na analogový signál zajistí, že se původně digitální signál transformuje do podoby vhodné k přenosu analogovým prostředím. Při volbě typu modulace musíme zohlednit konkrétní podmínky přenosového média, především s ohledem na vnější rušivé elementy, a to jak přírodního (např. atmosférické výboje), tak umělého (např. průmyslové rušení) původu.

1.1.3.3 Chybovost komunikačního kanálu

Rádiový komunikační kanál je zatížen celou řadou vnějších negativních vlivů:

- šum,
- atmosférické poruchy,
- průmyslové rušení,
- mnohocestné šíření, odrazy, úniky,
- záměrné rušení.

Z výše popsaných důvodů dochází při přenosu informací k chybám. Parametr, který popisuje úroveň chybovosti přenosového kanálu, se nazývá **BER** a je definován jako poměr chybně přijatých bitů vůči celkovému počtu všech přijatých bitů za určitý časový interval [2]:

$$BER = \frac{n_{chyba}}{n_{celkem}} \tag{1.3}$$

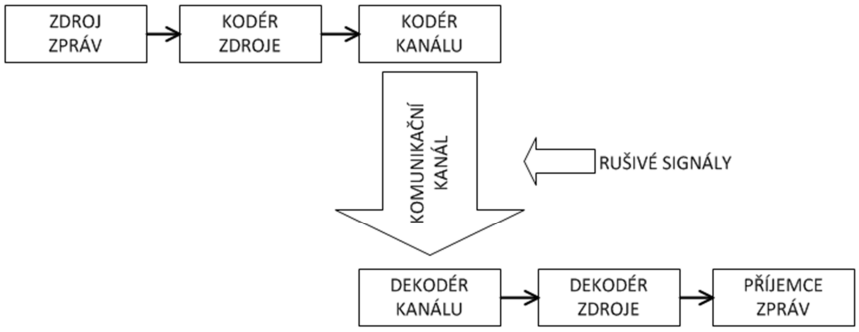
Různé systémy přenosu informací mají odlišné požadavky na maximální hodnotu BER. Například vojenské digitální polní telefonní ústředny vyžadují pro bezproblémovou synchronizaci BER maximálně 10E-4, digitální televize ve vysokém rozlišení pak BER 10E-10.

Chyby dále dělíme podle jejich výskytu na náhodné a shlukové [1]. Náhodné chyby jsou rozloženy přibližně rovnoměrně přes celou přenášenou zprávu a jsou na sobě nezávislé. Příčinou je obvykle šum. Shlukové chyby se projevují jako série po sobě se vyskytujících chyb v jedné části zprávy. Jsou způsobeny např. průmyslovým rušením (impulzní poruchy). Z hlediska detekce a následné korekce jsou pro nás příznivější chyby náhodné, které můžeme korigovat kanálovými zabezpečovacími kódy. V případě shlukových chyb pak musíme přikročit ještě k dalším mechanismům, například prokládání.

1.1.4 Kódování

Kódování obecně zajišťuje transformaci zprávy do takového formátu, aby ji bylo možné doručit příjemci dostupným komunikačním kanálem. Dále musíme počítat s vnějšími rušivými vlivy, které budou na kanál působit. Vznikají nám dvě oblasti problematiky kódování [1]:

- **Kódování zdroje:** zajišťuje přizpůsobení zdroje zprávy na dostupný komunikační kanál,
- **Kódování kanálu:** používá se pro potlačení negativních vlivů, působících v přenosovém kanále. Může se jednat například o proti-chybové kódování.



Obrázek 1.3: *Komunikační řetězec doplněný kodéry a dekodéry (převzato z [1])*

1.1.4.1 ***Kodér zdroje***

Jakákoliv informace, která má být přenesena digitálním komunikačním kanálem, musí být nejprve převedena na soustavu elektrických signálů. To je nejčastěji realizováno různými A/D převodníky neelektrických veličin na elektrické napětí. Takto získaná nová informace obvykle obsahuje velké množství redundance a irelevance, kterou je výhodné odstranit, z důvodu efektivního využití komunikačního řetězce. Proces potlačení nadbytečných informací se nazývá komprese a můžeme ji definovat následujícím vztahem [2]:

$$CR = \frac{v_{vst}}{v_{výst}} \tag{1.4}$$

Výsledkem je bezrozměrná veličina **kompresní poměr**, pomocí které dokážeme kvantitativně posoudit míru komprese. Proces komprese se příliš neprojeví u informací, které jsou v původní podobě generovány počítačem, protože mají již minimalizován výskyt redundance i irelevance [2].

Kompresi rozeznáváme dvojího druhu [2]:

- bezeztrátovou,
- ztrátovou.

Pomocí bezeztrátové komprese eliminujeme redundanci, která může být v informaci obsažena. Jedná se o proces vratný. Při rekonstrukci původního signálu v přijímači můžeme redundanci opětovně přidat, a tím získat původní zcela nezkreslenou informaci.

Naproti tomu ztrátová komprese odstraňuje složku irelevantní, která představuje nepodstatnou část informace. Problém nastává v okamžiku, kdy máme rozhodnout, co bude považováno za irelevantní, protože jakmile dojde jednou k eliminaci, již není možné tuto informaci v přijímači zpětně obnovit. Využíváme tak například nedokonalosti lidského zraku a odstraňujeme z obrazového signálu určité detaily, které objektivně snižují kvalitu obrazu, ale divák si takové změny ani nevšimne.

Typickými zástupci zdrojových kodérů jsou:

- kodéry tvaru vlny (PCM modulátory),
- parametrické kodéry (LPC kodéry),
- hybridní kodéry (LPC-RPE kodéry).

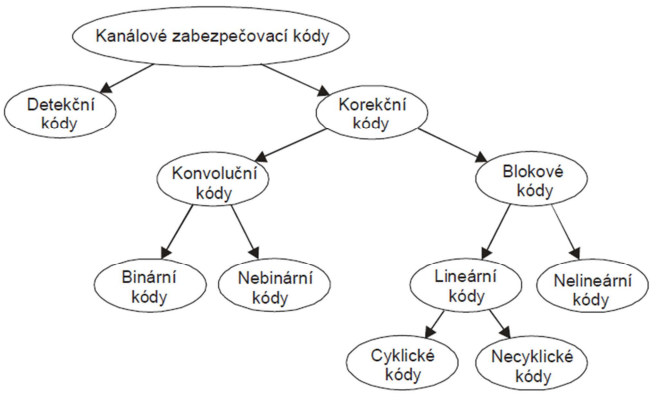
1.1.4.2 ***Kodér kanálu***

Úkolem kanálového kódování je přizpůsobit již digitalizovanou a nadbytečných informací prostou zprávu, pro přenos dostupným komunikačním kanálem. Především na bezdrátový komunikační kanál působí veliké množství rušivých vlivů, způsobených například šumem, odrazy, úniky apod. Zabezpečení zprávy dosáhneme opětovným přidáním redundance, tentokrát již ale přesně definované, která nám umožní detekovat, případně opravovat chyby, vzniklé během přenosu. Rychlost procesu kanálového kódování vyjádříme vztahem [2]:

$$R = \frac{n_{vst}}{n_{výst}} \tag{1.5}$$

Hodnota rychlosti kódování se pohybuje v rozmezí: $0 \leq R \leq 1$, kdy hodnota $R = 1$ znamená stav, kdy kanálové kódování není realizováno a přenos je zcela nechráněn, zatímco hodnota $R \rightarrow 0$ představuje stav dokonalé ochrany proti chybám, ale na úkor přenosové rychlosti užitečné informace, které by se také blížila k nule. Vhodně zvoleným kanálovým kódem dosáhneme výrazného snížení chybovosti komunikačního kanálu, za cenu jen mírného zvýšení redundance.

Rozdělení kanálových kódů ukazuje následující obrázek:



Obrázek 1.4: *Rozdělení kanálových kódů (převzato z [2])*

Kanálové kódy můžeme rozdělit podle toho, jak zabezpečí přenášené zprávy na dvě základní skupiny:

- detekční kódy,
- korekční kódy.

1.1.4.3 ***Detekční kódy***

Detekční kódy, jak už název napovídá, dokážou detekovat chybu, neumí ji však opravit. Pokud chceme dosáhnout efektivního a bezpečného přenosu informací, doplníme systém o zpětný kanál pro realizaci automatického opakování ARQ. Pokud kanálový dekodér přijímače vyhodnotí chybnou zprávu, vyšle zpětným kanálem žádost o opakování této konkrétní zprávy.

Nejjednodušším detekčním kódem je parita. Princip spočívá v přidání paritního bitu podle počtu jedniček obsažených ve zprávě. Parita může být buď sudá, anebo lichá. Pro zprávu 01001000 by byla sudá parita 1, lichá parita 0. Kontrola na straně přijímače je snadná a spočívá ve sčítání modulo 2 (XOR) bitů příchozí zprávy a porovnání výsledku s přijatou paritou. Paritní kód je jednoduchý na realizaci, velikou nevýhodu ale představuje skutečnost, že v případě dvojnásobné chyby je zpráva interpretována jako správná.

Často používaným detekčním kódem je CRC. Jedná se o způsob realizace kontrolního součtu nad vstupními daty. Výsledek je následně připojen k odesílané zprávě. Příjemce pak provádí vlastní výpočet nad přijatou zprávou a výsledek srovná s doručeným CRC součtem. Pokud jsou oba výsledky stejné, je došlá zpráva považována za bezchybnou, v opačném případě je prohlášena za neplatnou. CRC neumožňuje zjistit, kde k chybě došlo, ani její korekci. CRC kódy dělíme podle délky kontrolního součtu v bitech. Nejčastěji používané jsou CRC-8, CRC-16 a CRC-32. Podrobný princip výpočtu CRC je vysvětlen v kapitole 1.3.2.1.

1.1.4.4 ***Korekční kódy***

Korekční kódy naproti tomu umí chybu nejen detekovat, ale do určité míry i opravit. Takové systémy není nutné doplňovat o zpětný kanál, proto hovoříme o tzv. dopředné korekci chyb FEC. Tato skupina kódů patří mezi složitější a pro zabezpečení využívají větší počet bitů (redundance), než kódy detekční. Korekční kódy dále dělíme podle metody, jakou zpracovávají vstupní datový tok na konvoluční a blokové.

Konvoluční kodéry zpracovávají vstupní datový tok průběžně. Lze si je představit tak, že provádí konvoluci datového toku s impulzní odezvou kodéru. Rychlost kódování odpovídá vztahu 1.5. Příkladem konvolučního kódu je například Viterbiho algoritmus.

Blokové kodéry rozdělují vstupní datový tok do bloků o délce m bitů, ke kterým přidávají definovaný počet zabezpečujících bitů. Výsledný nový zabezpečený blok obsahuje n bitů. Počet zabezpečujících bitů je pak roven $k = n - m$. Kódy jsou značeny jako (n,m). Typickým zástupcem je rozšířený Hammingův kód (8,4), což znamená, že je na 4 vstupní datové bity použito 4 redundantních bitů a výsledný blok má celkovou délku 8 bitů. Rychlost kódování je v tomto případě $R = m/n = 4/8 = 0,5$. Hammingova vzdálenost, což je počet pozic o kolik se liší dvě slova stejné délky daného kódu, je $d = 4$, díky čemuž je možné pomocí tohoto kódu detekovat 2 chyby, nebo opravit jednu chybu. Konkrétní způsob realizace tohoto kódu je vysvětlen v kapitole 1.3.2.2.

1.1.5 **Šifrování**

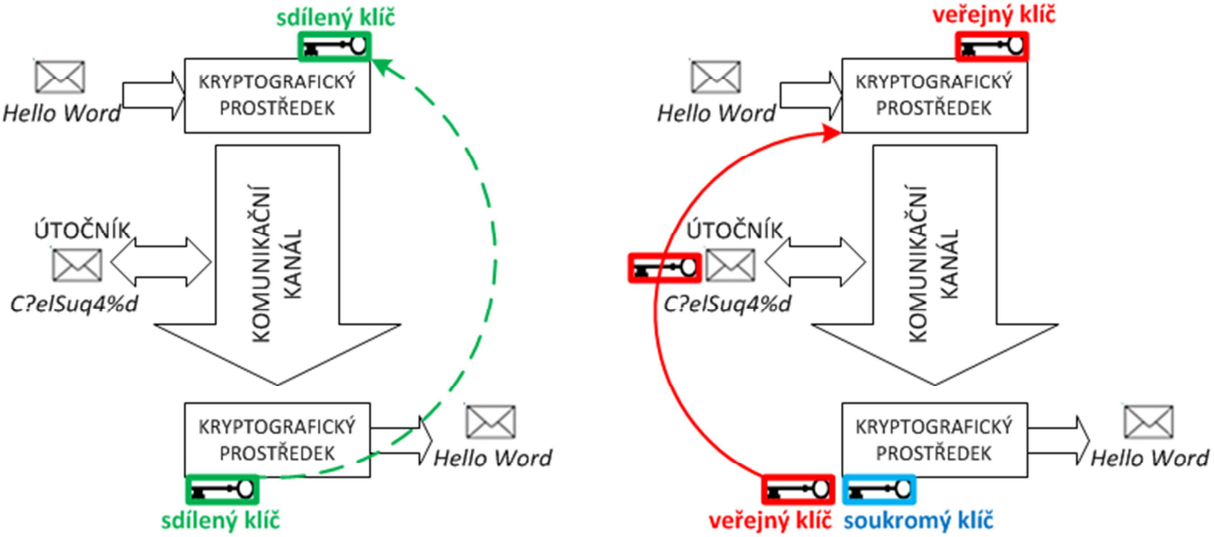
Velmi často dochází k přenosu citlivých informací, které jsou určeny pouze oprávněnému příjemci. Neustále musíme mít na paměti, že pokud není přenosový kanál fyzicky trvale střežen, může dojít k jeho narušení nezvaným útočníkem. Ten může provádět pouze pasivní odposlech, nebo se do komunikace aktivně zapojí pomocí podvržení vlastní identity (tzv. metoda „*man in the middle*“). V obou případech nemusíme být schopní narušení kanálu detekovat, a proto citlivé informace zabezpečujeme šifrovacími kódy. Informaci ze zašifrované zprávy dokáže získat jen oprávněný účastník vybavený správným klíčem.

Podle použití klíčů rozdělujeme kryptografické systémy na dva druhy:

- symetrické,
- nesymetrické.

U symetrických systémů je využito jediného klíče jak na straně odesílatele, tak i příjemce. Výhoda systému tkví v relativní jednoduchosti, ale velikou bezpečností trhlinou je fakt, že musíme zajistit bezpečný přenos klíče protistraně. V praxi nelze téměř 100% bezpečnost zajistit jinak, než pomocí kurýra, který fyzicky doručuje klíče oprávněným účastníkům, což je náročné na vynaložené prostředky a lidské zdroje.

Nesymetrická kryptografie odbourává nevýhodu předchozí varianty a to tak, že využívá dvou klíčů. Jeden privátní, který si příjemce uchovává v tajnosti a druhý veřejný. Ten může bez obav odeslat protistraně nezabezpečeným kanálem i s rizikem, že jej zachytí útočník. Protistrana zašifruje zprávu pomocí veřejného klíče a odešle ji příjemci. Zprávu lze zpětně dešifrovat pouze pomocí soukromého klíče, který ale vlastní jen příjemce. Útočník není schopen pomocí veřejného klíče šifrovanou zprávu rozkrýt. Schematicky jsou oba systémy ukázány na následujícím obrázku.



Obrázek 1.5: *Schéma symetrické a nesymetrické kryptografie*

V současné době existuje velké množství šifrovacích algoritmů využívajících jak symetrické, tak nesymetrické klíče. Ačkoliv je možné identifikovat lepší a horší varianty, volba konkrétního algoritmu vždy záleží na aplikaci, ve které chceme šifrování použít. Důležitá kritéria, která musíme zohlednit, jsou:

- kryptografická bezpečnost,
- složitost implementace,
- výkon.

Obecně platí, že snadno dosáhneme 2 ze 3 kritérií a posledního již obtížněji [12]. Výsledkem je kompromis, který nejlépe vyhoví zadaným požadavkům. Konkrétní šifrovací algoritmus použitý v této práci, je popsán v kapitole 1.4.

1.2 Kmitočtová pásma

Od počátků radiokomunikace uběhlo více jak jedno století. S tím, jak rostlo množství různých radiostanic, rostl i význam organizace radiového spojení v elektromagnetickém spektru. V současné době je používané radiové spektrum (9 kHz až 3000 GHz) považováno za velmi cenný prostor. Jeho rozdělení je přísně koordinováno v rámci mezinárodních a národních nařízení. Na nejvyšší úrovni je hlavním řídícím orgánem Mezinárodní telekomunikační unie ITU, která vydává Radiokomunikační řád. Ten je v České republice implementován prostřednictvím národní kmitočtové tabulky Českým telekomunikačním úřadem [4].

1.2.1 Kmitočtové pásmo ISM

V našem případě se budeme zajímat především o pásmo ISM, které je určeno pro průmyslové, vědecké a lékařské využití. Při dodržení podmínek stanovených ČTÚ je provoz zařízení v tomto pásmu bez poplatků, ale bez garance proti rušení. Pravidla použití upravuje všeobecné oprávnění ČTÚ č. VO-R/10/04.2012-7 k využívání rádiových kmitočtů a k provozování zařízení krátkého dosahu [7]. Následující tabulka ukazuje rozdělení kmitočtového pásma ISM pro nespecifikované stanice krátkého dosahu. Podrobný přehled je možné dohledat na webové adrese [6].

Tabulka 1.1: *Přehled pásma ISM*

Ozn.	Kmitočtové pásmo	Vyzářený výkon, popř. intenzita mag. pole	Kanálová rozteč	Klíčovací poměr
a	6765 - 6795 kHz	42 dBµA/m /10m	není stanovena	≤ 100 %
b	13,553 - 13,567 MHz	42 dBµA/m /10m	není stanovena	≤ 100 %
c	26,957 - 27,283 MHz	42 dBµA/m /10m nebo 10 mW e.r.p.	není stanovena	≤ 100 %
d	40,660 - 40,700 MHz	10 mW e.r.p.	není stanovena	≤ 100 %
e	138,200 - 138,450 MHz	10 mW e.r.p.	není stanovena	≤ 1 %
f	433,050 - 434,790 MHz	10 mW e.r.p.	není stanovena	≤ 10 %
f1	433,050 - 434,790 MHz	1 mW e.r.p.	není stanovena	≤ 100 %
f2	433,050 - 434,790 MHz	10 mW e.r.p.	max. 25 kHz	≤ 100 %
g	863,000 - 867,000 MHz	25 mW e.r.p.	max. 100 kHz	≤ 0,1 %
g1	868,000 - 868,600 MHz	25mW e.r.p.	není stanovena	≤ 1 %
g2	868,700 - 869,200 MHz	25mW e.r.p.	není stanovena	≤ 0,1 %
g3	869,300 - 869,400 MHz	25mW e.r.p.	max. 25 kHz	≤ 100 %
g4	869,400 - 869,650 MHz	500mW e.r.p.	max. 25 kHz	≤ 10 %
g5	869,700 - 870,000 MHz	5mW e.r.p.	není stanovena	≤ 100 %
g6	869,700 - 870,000 MHz	25mW e.r.p.	není stanovena	≤ 100 %
h	2400,0 - 2483,5 MHz	25mW e.r.p.	není stanovena	≤ 100 %
i	5725 - 5875 MHz	25mW e.r.p.	není stanovena	≤ 100 %
j	24,000 - 24,250 GHz	100mW e.r.p.	není stanovena	≤ 100 %
k	61,0 - 61,5 GHz	100mW e.r.p.	není stanovena	≤ 100 %
l	122 - 123 GHz	100mW e.r.p.	není stanovena	≤ 100 %
m	244 - 246 GHz	100mW e.r.p.	není stanovena	≤ 100 %

Kmitočty jsou určeny zejména pro telemetrii, dálkové ovládání, signalizaci a přenos poplachových informací. V našem případě budeme uvažovat, vzhledem k použitým modulům, pásmo s označením f2.

Vidíme ale, že je možné aplikovat bezdrátový přenos i pro diametrálně odlišná pásma. V každém případě ale musíme dodržet omezení týkající se výkonu, kanálové rozteče a klíčovacího poměru, abychom maximálně omezili negativní dopady rušení na okolní radiostanice, které pracují ve stejném pásmu.

1.3 Zabezpečení rádiového přenosu proti chybám

Jak už bylo naznačeno v kapitole 1.1.3.3, je reálný rádiový komunikační kanál zatížen chybami, které podle svého charakteru zasahují jednu nebo více částí přenášené zprávy. Pokud bychom přenášeli zprávy bez jakékoliv dodatečné ochrany, vystavujeme se vysokému riziku chybných přenosů, a tím pádem i nefunkčnosti spojení. Toto riziko je zvlášť vysoké u datových spojení, kde je nutné pro správnou interpretaci přenést data bezchybně, na rozdíl například od fonického spojení, kdy řeč obsahuje poměrně vysokou míru redundance a tak si příjemce dokáže chybějící část zprávy případně domyslet z kontextu. Celý problém rušení je dále prohlouben volbou kmitočtového pásma, kdy v ideálním případě máme některé kmitočty v daný čas a na konkrétním místě, vyhrazeny jen pro sebe. Naopak použití bez licenčního pásma sebou nese riziko, že v určené oblasti nebudeme sami a tak nám mimo přirozené rušení ještě přibude rušení od ostatních vysílačů, jejichž provoz není téměř nijak regulován.

Z nastíněných komplikací a vzhledem k uvažované aplikaci (datové přenosy), nám vyplývá nutnost zabezpečit přenášené zprávy proti chybám. Z rozboru v kapitole 1.1.4 vidíme, že existují v zásadě dvě možnosti:

- detekční kód se zpětným kanálem ARQ,
- korekční kód s dopřednou korekcí FEC.

1.3.1 Volba zabezpečovacích kódů

1.3.1.1 Volba detekčního kódu

Nejjednodušší detekční kód představuje parita. Její princip, výhody a nevýhody, již byly popsány v kapitole 1.1.4.3 a dále se s ním zabývat nebudeme, pro jeho slabou detekční schopnost. Mnohem zajímavější variantu představuje CRC a to především díky své nepoměrně vyšší detekční schopnosti a poměrně snadné implementaci.

U CRC kódu musíme zohlednit dvě hlavní kritéria:

- délku kontrolního součtu,
- volbu generujícího polynomu.

První parametr udává, kolik bitů bude mít výsledná kontrolní suma. Větší délka má lepší detekční vlastnosti, ale zase naroste přidaná redundance. Detekční schopnost CRC je určena dle vztahu:

$$d_{CRCn} = 1 - \frac{1}{2^n} \tag{1.6}$$

U CRC s délkou 8 bitů je detekční schopnost dle vztahu (1.6) rovna 0,99609, pro CRC s 16 bity pak 0,99998. Pro naše účely postačí délka 8 bitů. Zamýšlenému radiovému paketu délky 16 bajtů naroste redundance o 6,25%, přičemž výměnou získáme vysokou míru detekce chybných dat.

Druhým parametrem, který musíme u CRC zvolit, je generující polynom *g(x)*. Studiem problému zjistíme, že polynomy pro různé délky CRC jsou již určeny a standardizovány. Tabulka 1.2 představuje varianty generujících polynomů pro CRC-8. Pro snadnější orientaci zápis polynomu respektuje jak běžnou konvenci (vynechání MSB), tak i styl zápisu použitý v [8], kde autor naopak vynechává LSB.

Tabulka 1.2: *Generující polynomy pro CRC-8*

Název CRC8	**hex**	**Koopman**	**Polynom**
CCITT	0x07	0x83	$x^8 + x^2 + x + 1$
Dallas/Maxim	0x31	0x98	$x^8 + x^5 + x^4 + 1$
CRC8	0xD5	0xEA	$x^8 + x^7 + x^6 + x^4 + x^2 + 1$
SAE J1850	0x1D	0x8E	$x^8 + x^4 + x^3 + x^2 + 1$
WCDMA	0x9B	0xCD	$x^8 + x^7 + x^4 + x^3 + \mathrm{x} + 1$
DARC8	0x39	0x9C	$x^8 + x^5 + x^4 + x^3 + 1$
C2	0x2F	0x97	$x^8 + x^5 + x^3 + x^2 + \mathrm{x} + 1$
"*new*"	0x4D	0xA6	$x^8 + x^6 + x^3 + x^2 + 1$

S přihlédnutím k [8], kde je v tabulce 3 uveden přehled "nejlepších" polynomů s ohledem na délku kódované zprávy, nám jako nejlepší použitelný polynom vychází WCDMA.

1.3.1.2 ***Volba korekčního kódu***

Korekčních kódů existuje celá řada, opět se liší především svou schopností detekovat, opravovat chyby a složitostí implementace. Pro naše účely rádiového přenosu na krátké vzdálenosti se jeví jako ideální rozšířený Hammingův kód (8,4), a to především díky relativně snadné implementaci a dostatečné detekční a korekční schopnosti [9].

V případě výskytu chyby v jednom bitu přenášeného bajtu bude provedena automatická detekce a korekce. Pokud se chyba objeví na dvou bitech současně, je kód schopen toto identifikovat a přenesený bajt prohlásit za vadný. Kvůli absenci zpětného kanálu budou data nenávratně ztracena.

Hlavní výhodou systému je, jak už bylo naznačeno, možnost do určité míry zabezpečit přenos zpráv bez použití zpětného kanálu. Nevýhoda zabezpečení plyne ze samotné podstaty většího přidávání redundance oproti detekčním kódům. V našem případě připadnou na 4 informační bity 4 zabezpečovací bity. Důsledkem je, že musíme každý informační bajt rozdělit do dvou rádiových bajtů, a tím nám klesne přenosová kapacita komunikačního kanálu na polovinu.

1.3.2 **Algoritmus výpočtu**

1.3.2.1 ***Výpočet detekčního CRC kódu***

Vlastní CRC je reprezentován jako zbytek po dělení vstupní datové zprávy *M(x)* a generujícího polynomu *g(x)*. V oblasti počítačového zpracování pak s výhodou využíváme logické funkce XOR a posuvného registru, přes který postupně prochází vstupní datová zpráva *M(x)*.

Vstupní datová zpráva $M(x)$: "W" = 0x57 = 0b0101 0111
Generující polynom $g(x)$: $x^3 + x^2$ = 0b(1)1100

Výpočet:

```
        0101 0111|0000
XOR      111 00
        ---------------
        0010 0111|0000
XOR        11 100
        ---------------
        0001 1111|0000
XOR         1 1100
        ---------------
        0000 0011|0000
XOR              11 100
        ---------------
        0000 0000|1000
```

Obrázek 1.6: *Postup výpočtu CRC*

Výsledkem CRC-4 vstupní zprávy ASCII "W" při zadaném polynomu je číslo 0x8. Takto vzniklý CRC, coby redundanci, připojíme k původní datové zprávě a odešleme příjemci. Ten provede stejný výpočet a srovná výsledky. Pokud se budou shodovat, prohlásí přijatou zprávu za platnou. V případě, že se výsledky nebudou shodovat, tak nejprve došlou zprávu zneplatní a následně pomocí druhého mechanismu, kterým je zpětný kanál, odešle žádost odesílateli o opakování zprávy a celý postup se opakuje až do doby, kdy je zpráva doručena v pořádku a výsledky CRC se shodují.

1.3.2.2 ***Výpočet korekčního Hammingova kódu (8,4)***

Výpočet Hammingova kódu je založen na principu počítání parit ve vstupní datové zprávě. Nejprve rozdělíme datový bajt na dva půl-bajty. Následně z datových bitů vypočítáme podle zadaného algoritmu jednotlivé paritní bity, a ty připojíme za datový půl-bajt. Při výpočtu, podobně jako v předchozím případě, vhodně využíváme logické funkce XOR. Nově vzniklý bajt je nakonec odeslán protistraně.

A) KÓDOVÁNÍ
Vstupní datová zpráva: "W" = 0x57 = 0b0101 0111
Horní půlbajt: 0b0101 (datové bity: D_0, D_1, D_2, D_3)

Úprava půlbajtu pro umístění parit: 0101 $\underline{\ }\ \underline{\ }\ \underline{\ }\ \underline{\ }$
$P_1\ P_2\ P_4\ P_8$

Výpočet paritních bitů:
$P_1 = D_0 \oplus D_1 \oplus D_2 = 0 \oplus 1 \oplus 0 = \mathbf{1}$
$P_2 = D_0 \oplus D_2 \oplus D_3 = 0 \oplus 0 \oplus 1 = \mathbf{1}$
$P_4 = D_0 \oplus D_1 \oplus D_3 = 0 \oplus 1 \oplus 1 = \mathbf{0}$
$P_8 = D_0 \oplus D_1 \oplus D_2 \oplus D_3 \oplus P_1 \oplus P_2 \oplus P_4 = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = \mathbf{0}$

Výsledná zabezpečená zpráva: 0b0101 **1100**

B) DEKÓDOVÁNÍ
Výpočet syndromu:
$S_1 = P_1 \oplus D_0 \oplus D_1 \oplus D_2 = 1 \oplus 0 \oplus 1 \oplus 0 = \mathbf{0}$
$S_2 = P_2 \oplus D_0 \oplus D_2 \oplus D_3 = 1 \oplus 0 \oplus 0 \oplus 1 = \mathbf{0}$
$S_4 = P_4 \oplus D_0 \oplus D_1 \oplus D_3 = 0 \oplus 0 \oplus 1 \oplus 1 = \mathbf{0}$
$S_8 = P_8 \oplus D_0 \oplus D_1 \oplus D_2 \oplus D_3 \oplus P_1 \oplus P_2 \oplus P_4 = 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = \mathbf{0}$

syndrom: 0b0000 = 0 => data bez chyb

Obrázek 1.7: *Postup výpočtu Hammingova kódu (8,4)*

Ve výsledné zprávě vidíme vložené zabezpečovací bity. Příjemci jsou polohy datových a paritních bitů předem známy a po přijetí provede stejný výpočet s tím rozdílem, že k nově počítaným paritám zahrne také přijaté kontrolní bity. Výsledek výpočtu všech parit (tzv. syndrom) musí být 0. V případě, že syndrom vyšel nenulový, podívá se příjemce do look-up tabulky, kde jsou různým hodnotám syndromu přiřazeny jednotlivé bitové pozice přijatého slova. Pokud najde odpovídající hodnotu syndromu v tabulce, provede automaticky korekci příslušné bitové pozice pomocí logické funkce negace a přijatý bajt je prohlášen za platný. V případě že se hodnota syndromu v tabulce nenalézá, došlo při přenosu minimálně ke dvojnásobné chybě. Takovou již algoritmus neumí korigovat, ale označí přijatý bajt za neplatný.

Tabulka 1.3: *Look-up tabulka Hammingova kódu (8,4)*

syndrom (hex)	**bitová pozice (zprava)**
0x8	0 (LSB)
0xC	1
0xA	2
0x9	3
0xB	4
0xD	5
0xE	6
0xF	7 (MSB)

1.3.3 Softwarová implementace

1.3.3.1 Softwarová implementace CRC kódu

Výpočet CRC lze v zásadě implementovat dvěma způsoby, lišícími se spotřebou paměti a výpočetní náročností.

- Univerzálně (nižší nároky na paměť, vyšší výpočetní náročnost),
- pomocí look-up tabulky (vyšší nároky na paměť, menší výpočetní náročnost).

Univerzální přístup spočívá v tom, že funkce, realizující CRC výpočet přímo provádí klasické dělení polynomů. Výhodou této varianty je, že můžeme libovolně měnit generující polynom a také nízké paměťové nároky. Nevýhoda je zřejmá. Výpočet trvá poměrně dlouhou dobu, což je nežádoucí, aby nám výpočet na MCU tvořil "úzké hrdlo" celého komunikačního řetězce.

Druhý přístup využívá look-up tabulky, která je pro každý generující polynom jiná. V případě CRC-8 zabírá v paměti 256 bajtů. V tabulce jsou uloženy před vypočítané zbytky dělení pro všech 256 čísel 8 bitového slova. Samotný algoritmus ve výpočtu používá index, na základě něhož se podívá do look-up tabulky a ihned přečte zbytek po dělení. V samotné realizaci tabulky máme ještě 2 možnosti.

Inicializovat tabulku pomocí funkce po zapnutí systému. To přinese opět výhodu, podobně jako u univerzálního přístupu, kdy můžeme libovolně změnit polynom a tabulka se vždy přepočítá znova. Problém nastává u zamýšlené architektury 8 bitového MCU Atmel AVR. Takto inicializovaná tabulka by musela být umístěna v paměti RAM, která je ale řádově menší než paměť FLASH a zároveň zabírá poměrně dost místa.

Druhou možností je tabulku vypočítat pro jeden konkrétní generující polynom (např. pomocí některého z on-line kalkulátorů) a uložit ji při tvorbě program do paměti FLASH. Zkombinujeme tím výhodu rychlého výpočtu a zároveň částečně vyřešíme paměťový problém při zachování vysoké rychlosti přístupu k tabulce. Jedinou nevýhodou tak zůstává nemožnost změnit generující polynom za běhu programu. To ale částečně korigujeme výběrem nejlepšího dostupného polynomu dle [8].

1.3.3.2 Softwarová implementace Hammingova kódu (8,4)

V programovacím jazyce C lze výpočet Hammingova kódu (8,4) implementovat snadno. Pro architekturu Atmel AVR využíváme s výhodou datový typ "*bit*", který je realizován překladačem v poli univerzálních I/O registrů. Díky němu můžeme pracovat s bajtem na úrovni jednotlivých bitů. Při dekódování nám zase pomůže přepínač "*switch*", díky kterému zrealizujeme look-up tabulku. Celý algoritmus je poměrně nenáročný na paměť (FLASH i RAM) i výpočetní výkon, což je zvláště výhodné u zvolené architektury.

1.4 Zabezpečení rádiového přenosu proti odposlechu

Hledisko bezpečnosti přenosu informací proti náhodnému nebo úmyslnému odposlechu, je v současné době stále diskutovanější téma. Čím více dnes využíváme prostředků elektronické komunikace, tím více informací v případě nulového (nebo nevyhovujícího) šifrování o nás útočník získá. Jednotlivé technické prostředky jsou v podstatě předurčeny pro konkrétní typ komunikace a tomu by mělo odpovídat i adekvátní zabezpečení dat. V bez licenčním pásmu 433 MHz předpokládáme výskyt především automatických stanic krátkého dosahu přenášejících povelové nebo telemetrické zprávy nízkého stupně důležitosti. Tím nám nevzniká akutní potřeba šifrování.

Můžeme ale například navrhnout zabezpečovací systém rozsáhlého pozemku, jehož čidla budou komunikovat s řídící jednotkou bezdrátově. Tady už šifrování nabývá na významu, protože útočník nebude schopen rozkrýt komunikaci (nebo jen s použitím nepřiměřených nákladů) a případně podvrhnout identitu jednotlivých čidel, aby následně vnikl na pozemek.

1.4.1 Srovnání šifrovacích algoritmů

Šifrovacích algoritmů a principů činnosti existuje veliké množství. Vzhledem k použité architektuře řídícího systému radiostanice - 8 bitový MCU bez hardwarově vestavěného šifrovacího modulu, bylo nutné zvolit algoritmus, který je co nejjednodušší na softwarovou implementaci a poskytuje maximální možný výkon při spotřebě co nejmenšího množství omezených systémových prostředků (RAM, CPU, FLASH). Hledisko bezpečnosti je až na druhém místě, právě kvůli již zmíněnému předpokládanému využití systému.

Následující tabulka je převzata z [12], a ukazuje srovnání v současnosti často používaných algoritmů.

Tabulka 1.4: *Srovnání šifrovacích algoritmů*

Algoritmus	8 bit prostředí (C, asm)	32 bit prostředí (C)	64 bit prostředí (C, asm)	Obtížnost implementace	Celková bezpečnost	CELKEM
MARS	2	2	2	1	3	**10**
RC6	2	3	2	1	2	**10**
Rijndael	3	2	3	3	2	**13**
Serpent	1	1	1	3	3	**9**
Twofish	2	2	3	2	3	**12**

Hodnoty jsou relativní k ostatním uvedeným algoritmům pro 128 bitové klíče a platí, že vyšší číslo znamená lepší výsledek. Z tabulky vidíme, že celkově nejlepší algoritmus je Rijndael, v těsném závěsu za ním pak Twofish, který ale poskytuje vyšší bezpečnost. Na začátku jsme si definovali, že hledisko výkonu a snadné implementace má přednost před bezpečností. Pro další úvahy tak budeme nadále uvažovat už jen o algoritmu Rijndael.

1.4.2 Algoritmus AES

Původním názvem "Rijndael" je standardizovaný šifrovací algoritmus schválený Americkým úřadem pro standardizaci v roce 2001. Vznikl z potřeby nahradit předchozí šifrovací systém DES, respektive bezpečnější variantu 3DES, která ale nedosahovala požadovaného výkonu [12].

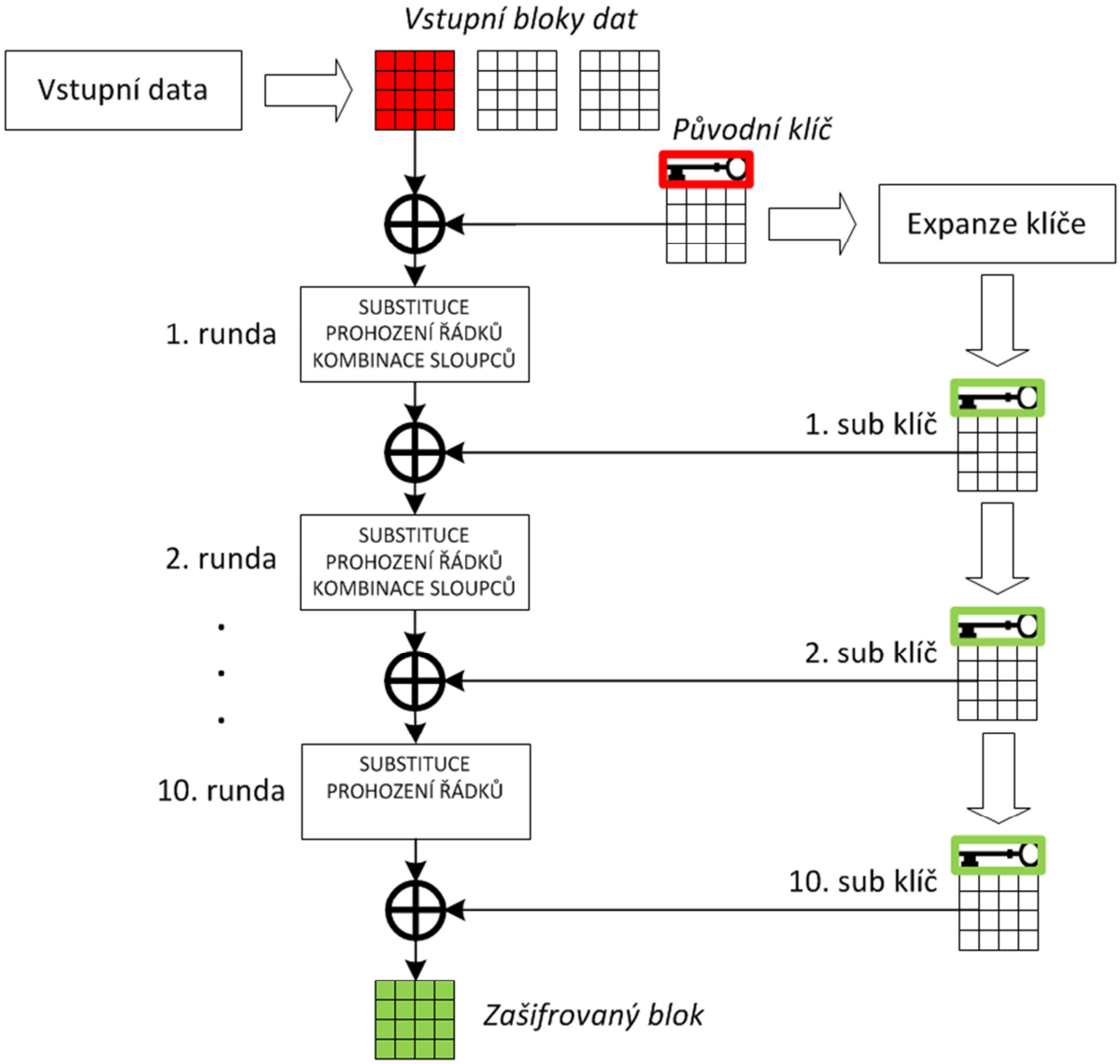
Jedná se blokovou symetrickou šifru s blokem délky 128 bitů a klíčem délky 128,192 nebo 256 bitů. Bloky mají délku vždy fixní, délku klíče je možné zvolit podle potřeby. Platí, že delší klíč přináší vyšší úroveň bezpečnosti. Konkrétní algoritmus výpočtu je zcela otevřený a kdokoliv, včetně komerčních organizací jej může využít. To představuje velikou výhodu při hledání optimalizovaného kódu pro konkrétní platformu.

Nevýhodu představuje, jak už z popisu vyplývá, právě použití sdíleného klíče pro symetrickou kryptografii. Musíme zajistit bezpečný přenos klíče příjemci zprávy tak, aby nemohl být zachycen útočníkem. Pro naši aplikaci to nepředstavuje až tak významný problém, protože předpokládáme vzdálenost radiostanic maximálně v řádu stovek metrů, a tudíž můžeme nastavit klíč na každé stanici osobně, aniž bychom ho byli nuceni posílat. Tohoto faktu je také využito například u zabezpečení bezdrátových Wi-Fi sítí, kde je AES implementován jako součást zabezpečení WPA2 [11].

1.4.3 Princip činnosti AES

Následující postup objasňuje princip činnosti AES při použití 128 bitového klíče [10]:

- V prvním kroku jsou vstupní data rozdělena do bloků o velikosti 128 bitů (matice 4x4 bajty) a v úvodní rundě jsou tato data XORována s původním klíčem.
- V dalším kroku dochází k expanzi klíče, kdy se v původním klíči vybírají jednotlivé sloupce, mění pořadí řádků, a ty dále prochází substitucí přes tzv. s-box, který původní bajty nahrazuje novými. Nově vzniklé sloupce jsou dále XORovány s rundovní konstantou. Výsledek je pak ještě jednou XORován s odpovídajícím sloupcem klíče z předchozí rundy. Takto je postupně za sebou sestaveno 10 nových sub klíčů (pro 128 bitový klíč).
- Následně jsou v 10ti rundách provedeny nad maticí s daty operace substituce (původní bajty jsou nahrazeny novými na základě průchodu s-boxem), prohození řádků, kombinace bajtů ve sloupci a nakonec XOR s příslušným rundovním klíčem.
- V 10. rundě se neprovede kombinování bajtů ve sloupci.
- Výsledkem průchodu 10. rundou je zašifrovaný blok dat.
- Dešifrování probíhá v opačném pořadí.



Obrázek 1.8: *Šifrovací algoritmus AES*

1.5 Režie radiového spojení

Vzhledem k popsaným možnostem zabezpečení radiového přenosu proti chybám a odposlechu, je nutné nějakým způsobem řídit radiový přenos tak, aby přijímač vždy korektně rozeznal, jaký režim zabezpečení byl použit. Dále přijímací strana nikdy dopředu neví, kdy a jaké množství informací budou přenášeny, a proto musí umět bezpečně rozeznat ukončení radiové relace.

Inspiraci k řešení problému můžeme nalézt u již existujících používaných přenosových protokolů (např. protokol TCP/IP), kde hlavní filosofie spočívá v tom, že data jsou rozdělena do krátkých úseků (paketů), z nichž každý je doplněn hlavičkou, která popisuje vlastnosti konkrétního paketu, a zabezpečen některou z metod kontrolního součtu. Pro náš případ použijeme stejný systém, jen upravený pro konkrétní aplikaci.

1.5.1 Režimy spojení

V souladu se zadáním a podle rozboru provedeného výše, definujeme jednotlivé režimy spojení. Uvažujeme jak situace, kdy je spojení zcela bezchybné, tak případy, kdy dojde k rušení radiového kanálu. Tabulka ukazuje možnosti zabezpečení pro různé aplikace, které budou v radiostanici obsaženy.

Tabulka 1.5: *Kombinace režimů spojení*

aplikace	**zabezpečení proti chybám**			**zabezpečení proti odposlechu**
	bez zabezpečení	FEC	ARQ	AES
PING test	✗	✗	✗	✗
Měření chybovosti rád. kanálu	✓	✓	✗	✗
Přenos textu	✓	✓	✓	✓
Přenos dat	✓	✓	✓	✓

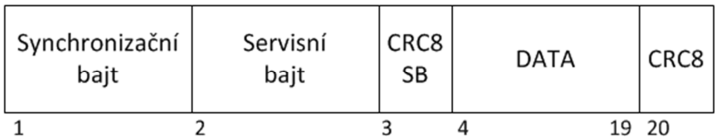
1.5.2 Struktura (délka) datového paketu

Spojení v ISM pásmu 433 MHz je, jak bylo zmíněno v kapitole 1.2, prakticky nijak neregulováno, a proto očekáváme poměrně silné rušení. Z tohoto důvodů budeme volit délku paketu co nejkratší, abychom maximálně omezili čas potřebný k přenosu paketu.

Při návrhu délky datové části vycházíme z požadavků šifrovacího algoritmu AES, který pracuje s 16 bajtovými bloky. Pro minimální délku paketu, který umožní pohodlné šifrování a dešifrování proto volíme právě 16 bajtů. Datové pole, ať už šifrované nebo ne, musíme chránit proti chybám vzniklým během přenosu. K tomu využijeme algoritmus CRC-8, jehož výsledek v podobě jednoho bajtu přidáme za datový blok. Nová délka paketu je 17 bajtů.

Pro popis vlastností paketu použijeme jeden bajt, který bude dále nazýván jako **servisní bajt**. Ten nám umožní definovat až 256 vlastností konkrétního paketu (tj. jaký typ zabezpečení proti chybám je aplikován, jestli jsou data šifrována, případně jestli se nejedná o některý ze speciálních paketů) a díky němu může přijímač provést odpovídající operace. Servisní bajt je pro korektní datové spojení naprosto klíčovým prvkem, a pokud by byl přenesen s chybou, musí přijímač tuto skutečnost bezpečně odhalit a raději celý paket zahodit. Z tohoto důvodu bude sám zabezpečen vlastním kontrolním součtem CRC8 v podobě dalšího bajtu. Délka paketu nám tak narostla již na 19 bajtů.

Poslední bajt, který ještě přidáme, vychází z empirického pozorování, kdy bylo zjištěno, že radiové moduly provozované v pásmu 433 MHz velmi často indikují falešný příjem. Zdrojem takovýchto rušení jsou různá okolní zařízení (termostaty, meteostanice). Zamýšlený bajt bude mít konstantní hodnotu, kterou získáme experimentálně a bude nastavena tak, aby docházelo co nejméně k falešným příjmům. Celková délka paketu nám narostla na výsledných 20 bajtů. Grafické znázornění ukazuje následující obrázek:



Obrázek 1.9: *Struktura datového paketu*

1.5.3 Formát hlavičky

Za hlavičku můžeme v našem případě považovat první tři bajty. Hodnota synchronizačního bajtu bude vždy nastavena na 0xCA. Hodnoty servisního bajtu určíme tak, že si do tabulky sepíšeme všechny zamýšlené provozní režimy radiostanice a k nim doplníme konkrétní hodnoty, a to buď "libovolně" nebo podle nějakého klíče. V našem případě je tabulka následující:

Tabulka 1.6: *Hodnoty servisního bajtu*

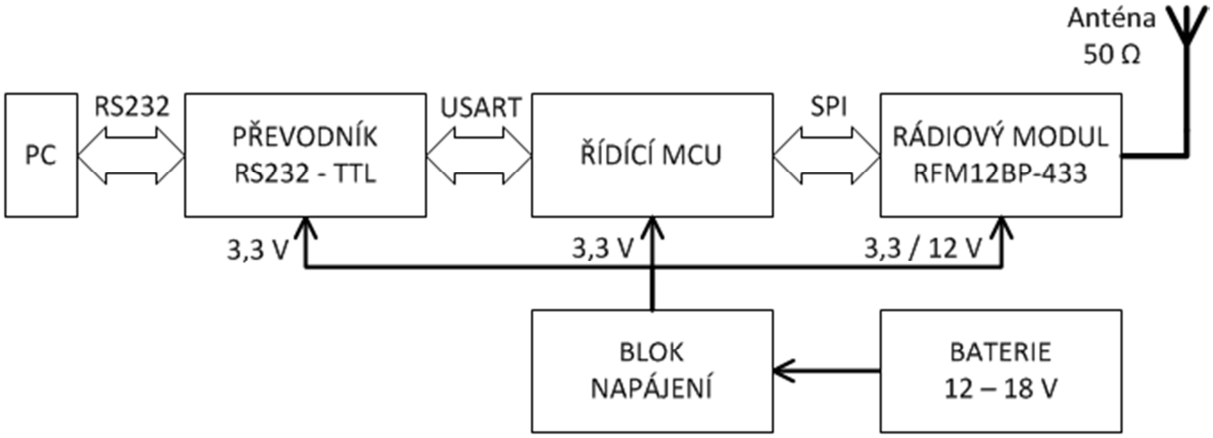
Význam SB	Hex	CRC8
řízení spojení		
odpověď ARQ OK	0x28	0x7D
PING dotaz	0x48	0xD1
PING odpověď	0x40	0xC8
SERIAL MODEM - připraven na příjem dat	0xF0	0x95
SERIAL MODEM - délka datového proudu	0xF1	0xE0
SERIAL MODEM - konec souboru (režim normal a FEC)	0xFD	0xD6
SERIAL MODEM - konec souboru (režim ARQ)	0xFF	0x7B
měření chybovosti radiového kanálu		
BER TEST - bez ochrany přenosu	0x08	0x19
BER TEST - zabezpečení FEC - 1. paket	0x0C	0xD8
BER TEST - zabezpečení FEC - 2. paket	0x0E	0x75
přenos textových zpráv		
TEXT - bez ochrany přenosu dat, bez šifrování	0x88	0x12
TEXT - bez ochrany dat, AES	0x89	0x89
TEXT - ARQ, bez šifrování - 1. pokus	0x8A	0xBF
TEXT - ARQ, AES - 1. pokus	0x8B	0x24
TEXT - ARQ, bez šifrování - 2. pokus	0x9A	0x8D
TEXT - ARQ, AES - 2. pokus	0x9B	0x16
TEXT - FEC, bez šifrování - 1. paket	0x8C	0xD3
TEXT - FEC, bez šifrování - 2. paket	0x8E	0x7E
TEXT - FEC, AES - 1. paket	0x8D	0x48
TEXT - FEC, AES - 2. paket	0x8F	0xE5
přenos dat		
SERIAL MODEM - bez ochrany přenosu dat, bez šifrování	0xC8	0xDA
SERIAL MODEM - bez ochrany dat, AES	0xC9	0x41
SERIAL MODEM - ARQ, bez šifrování - 1. pokus	0xCA	0x77
SERIAL MODEM - ARQ, AES - 1. pokus	0xCB	0xEC
SERIAL MODEM - ARQ, bez šifrování - 2. pokus	0xDA	0x45
SERIAL MODEM - ARQ, AES - 2. pokus	0xDB	0xDE
SERIAL MODEM - FEC, bez šifrování - 1. paket	0xCC	0x1B
SERIAL MODEM - FEC, bez šifrování - 2. paket	0xCE	0xB6
SERIAL MODEM - FEC, AES - 1. paket	0xCD	0x80
SERIAL MODEM - FEC, AES - 2. paket	0xCF	0x2D

Řídicí systém bude navržen tak, že přijímač nejprve načte celý nově příchozí paket, následně provede kontrolu servisního bajtu pomocí CRC-8 a v případě, že je bajt v pořádku, tak provede příslušnou operaci na základě jeho hodnoty. Tento koncept nám umožní zkonstruovat uživatelsky přívětivou radiostanici, kdy nebude bezpodmínečně nutné, aby se účastníci provozu dopředu domlouvali na parametrech zabezpečení radiového spojení. Bezpečnost bude plně v režii vysílací strany.

2 Praktická realizace radiostanice

2.1 Koncept radiostanice a použité komponenty

Blokové schéma radiostanice je uvedeno na následujícím obrázku.



Obrázek 2.1: *Blokové schéma radiostanice*

Hlavním řídícím prvkem je 8 bitový MCU Atmel AVR řady Mega. Komunikace s obsluhou je řešena přes standardní sériové rozhraní RS232, kdy je v roli převodníku napěťových úrovní použit oblíbený obvod MAX3232CPE od firmy Maxim. Rádiovou část stanice zajišťuje hotový modul RFM12BP-433 od firmy Hope Microelectronics, který se těší veliké oblibě jak u amatérských, tak komerčních projektů. Blok napájení je osazen dvěma integrovanými stabilizátory napětí. Prvním je typ LF33CV, který zajišťuje napětí 3,3V pro napájení logických obvodů a řídícího obvodu radiostanice. Druhý je pak tvořen nastavitelným stabilizátorem LM217T. Ten se stará o napájení koncového stupně vysokofrekvenčního zesilovače radiového modulu. Napětí je možno plynule nastavovat pomocí osazeného trimru. Na stanici jsou dále osazeny tři led diody připojené k MCU. Jejich hlavním účelem je indikovat jednotlivé stavy radiostanice. Dále nechybí dva mikrospínače. První provádí hardwarový reset MCU a druhý, označen jako "FACTORY DEFAULT", uvede radiostanici do tzv. továrního nastavení, které je uloženo v paměti FLASH. Tato funkce je obzvláště výhodná, pokud uživatel změní parametry komunikační linky RS232 a nedaří se mu opětovné spojení počítače s MCU. Nakonec je také osazeno ISP rozhraní pro možnost přeprogramování MCU bez nutnosti demontáže a dále jsou všechny porty MCU osazeny kolíkovými lištami pro snadné připojení dalších periferií nebo monitorovacích prostředků (logický analyzátor, osciloskop apod.).

Při volbě součástkové základny byl kladen důraz především na snadnou dostupnost v maloobchodní síti a následnou jednoduchost výroby v domácích podmínkách. Všechny součástky jsou v provedení pro THT montáž s roztečí 2,54 mm, kromě radiového modulu, pro který bylo nutné nejprve zhotovit adaptér, aby vývody dosáhly požadované rozteče.

2.1.1 Řídící MCU

Samostatnou kapitolou návrhu radiostanice byla volba vhodného řídícího MCU. Jak už bylo dříve naznačeno, jedná se o MUC od firmy Atmel. Výrobce byl zvolen již dopředu, a to především kvůli použitému programátoru Presto od firmy Asix, který velmi dobře podporuje programování MCU Atmel a Microchip. Dalším důležitým argumentem je využití výborného vývojového prostředí a překladače CodeVision AVR od firmy HP InfoTech. Ten je známý především propracovanou optimalizací a překladem programu v jazyce C pro architekturu Atmel AVR. Příjemným bonusem je dále kvalitně zpracovaná uživatelská dokumentace.

Kritéria, která vedla k výběru konkrétního modelu, byla následující (seřazeno podle priority od nejdůležitější po méně důležité):

- dostupnost v tuzemských obchodech,
- pouzdro DIL pro THT montáž s roztečí 2,54 mm,
- 8 bitová architektura,
- napájecí úroveň 3,3 V,
- paměť FLASH alespoň 32 kB,
- rozhraní USART,
- co největší paměť RAM,
- co nejvyšší taktovací kmitočet,
- alespoň dvě I/O linky s možností externího přerušení,
- podpora rozhraní SPI.

Po prozkoumání dostupných součástek na trhu, se objevila v podstatě jediná varianta, a to model Atmel AVR Mega 32A, který splňuje požadovaná kritéria. Konkrétní hlavní parametry jsou následující:

- Pouzdro: PDIP 40
- Napájecí napětí: 2,7 - 5,5 V
- Paměť FLASH: 32 kB
- Paměť RAM: 2 kB
- Paměť EEPROM: 1 kB
- Taktovací kmitočet: 16 MHz (pro Vcc = 4,5 V)
- Rozhraní: 1x SPI, 1x USART
- I/O linky s externím přerušením: 2x

Při návrhu zapojení musíme vzít v potaz souvislost mezi maximálním taktovacím kmitočtem a napájecím napětím. Graf závislosti je uveden v [13] na straně 298. Vidíme, že pro napájecí napětí 3,3 V můžeme použít krystal o maximálním kmitočtu přibližně 10,2 MHz, abychom se udrželi v oblasti, kdy výrobce ručí za vlastnosti MCU napříč teplotním rozsahem od -55°C do +125°C. Ačkoliv se při testování 16 MHz krystalu při pokojové teplotě a napětí 3,3 V neobjevovaly problémy, přesto raději použijeme "bezpečný" taktovací kmitočet pomocí krystalu 9,216 MHz, který je také výhodný pro časování USART komunikace.

2.1.2 Blok napájení

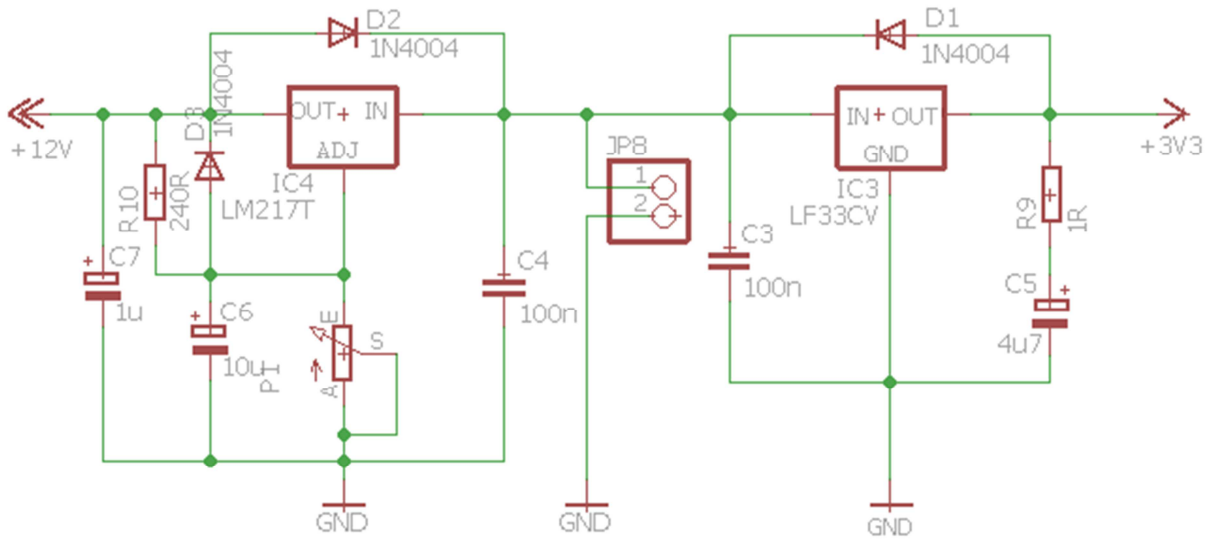
Z hlediska napájení je v konstrukci radiostanice nutné zabezpečit 3 hlavní požadavky:

- Stabilizované napájecí napětí 3,3 V s proudovou zatížitelností alespoň 500 mA.
- Stabilizované napájecí napětí 12 V s možností jemné regulace napětí a proudovou zatížitelností alespoň 500 mA.
- Celý blok musí být zapojen tak, aby jej bylo možné napájet z jediného zdroje (baterie) v širokém rozsahu napětí. To dovolí použít celou škálu zdrojů od běžných suchých monočlánků, přes moderní lithiové články až po olověné baterie osobních automobilů.

Zapojení stabilizátoru LF33CV vychází z obecných principů aplikace těchto obvodů. To znamená, že jsou umístěny dva blokovací kondenzátory, které vyhlazují napájecí napětí v případě náhlého zvýšení odběru proudu. Jeden na vstupu a druhý na výstupu. Dále je ke stabilizátoru doplněna běžná křemíková dioda 1N4004, která chrání stabilizátor proti působení zpětného proudu. Zapojení je ještě vylepšeno o rezistor s hodnotu 1 Ω, který je zapojen do série s kondenzátorem C5. Tato úprava vychází z informace uvedené v datasheetu [14], kde je u parametru "*Output Bypass Capacitance*" uvedena potřebná hodnota ESR v rozmezí 0,1 až 10 Ω. Pokud použijeme běžný elektrolytický kondenzátor, bude ESR s největší pravděpodobností v pořádku, problém by nastal např. u keramického kondenzátoru, který má ale ESR velmi malé. V tomto případě riskujeme, že při zapnutí napájení dojde k překmitu výstupního napětí a tím zničení součástek (MCU, převodník, řídící obvod radiového modulu). Přidaný rezistor nám zajistí za všech okolností bezpečnou hodnotu ESR a omezí riziko vzniku překmitu. Výsledný zdroj poskytuje proud až 1 A.

Regulovatelný stabilizátor LM217T byl zvolen z důvodů možnosti jemné regulace napájecího napětí koncového stupně VF zesilovače, a to proto, že při napájení 12 V dochází při nevhodně přizpůsobené anténě k občasným restartům radiového modulu. Zapojení kopíruje doporučení uvedené v katalogovém listu [15]. I zde vidíme aplikaci ochranných diod a blokovacích kondenzátorů. Díky tomu získáme při napájení 12 V baterií zdroj s možností regulace výstupního napětí v rozmezí 1,2 až 11,2 V a proudovou zatížitelností až 2,2 A.

Napájecí větev 3,3 V bude mít předpokládaný proudový odběr menší než 10 mA a na druhé větvi pro VF zesilovač je předpokládaný odběr do 200 mA v režimu vysílání. Díky takto malým odběrům nebude potřeba stabilizátory nijak dodatečně chladit, a to ani pasivním plechovým chladičem. Kompletní schéma zapojení napájecího bloku je ukázáno na následujícím obrázku.



Obrázek 2.2: *Schéma bloku napájení radiostanice*

2.1.3 **Převodník úrovní RS232 - TTL**

Požadavkem je, aby bylo možné ovládat radiostanici pomocí osobního počítače. Z důvodů jednoduchosti implementace bylo zvoleno sériové rozhraní RS232, jehož realizace je pro MCU Atmel AVR notoricky známá a velmi často používaná. Pro správné propojení sériové linky osobního počítače s vyššími napěťovými úrovněmi a použitého MCU je nutné použít odpovídající typ převodníku. Vzhledem k 3,3 V logice, byl zvolen integrovaný 4 kanálový převodník MAX 3232 CPE, který je dodáván v pouzdře DIP 16. Zapojení opět odpovídá doporučením uvedeným v katalogovém listu [16]. Radiostanice využívá pouze 2 kanály pro linky RXD a TXD. Řízení toku je realizováno softwarovou cestou mechanismem XON/XOFF.

Softwarová varianta řízení toku je zvolena z praktických důvodů. Moderní osobní počítače a především notebooky již nejsou běžně osazovány sériovým portem. Uživatelé jsou nuceni používat převodníky USB - RS232. Jejich ceny a kvalita provedení se však výrazně liší. U velmi levných modelů tak nemusí být zapojeny a podporovány linky používané pro hardwarové řízení toku.

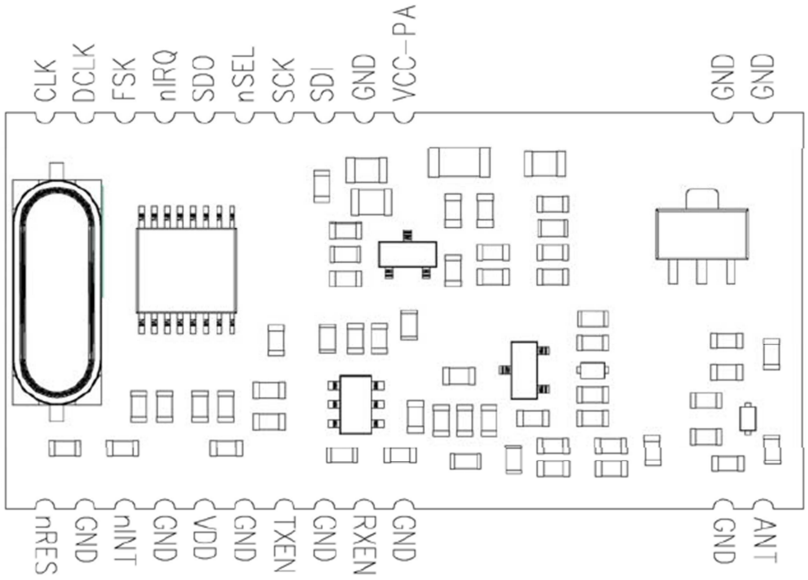
2.1.4 **Anténa**

Konkrétní provedení antény výrobce radiového modulu nepředepisuje. Musíme zabezpečit pouze to, aby byla anténa určena pro námi zvolené kmitočtové pásmo (433 MHz nebo 868 MHz). Anténa může být jak všesměrová (prutová, ground plane apod.), tak i směrová (YAGI), případně můžeme použít samotný koaxiální kabel s impedancí 50 Ω, který bude na konci zbaven stínění v délce odpovídající naladění na zvolené kmitočtové pásmo.

V případě, že budeme anténu sami vyrábět, tak máme naštěstí velké množství dostupných návodů jak v odborné literatuře [17], tak na internetu [18]. Dobrých výsledků, při zachování konstrukční jednoduchosti dosáhneme použitím 1/4 vlnné prutové antény.

2.2 Rádiový modulu RFM12BP-433

Pro realizaci zadání jsme zvolili rádiový modul RFM12BP ve variantě pro pásmo 433 MHz od čínské firmy Hope Microelectronics. Jedná se o malý transceiver, dobře dostupný v tuzemské obchodní síti s přijatelnými pořizovacími náklady. Mezi jeho hlavní přednosti patří integrovaný koncový stupeň VF zesilovače, široké možnosti nastavení parametrů a jednoduché připojení na řídící MCU, které je realizováno rozhraním SPI. Nevýhodou je poměrně zmatená dokumentace [19], [20], která je ale částečně suplována kvalitními zdroji na internetu [21], [22], [23].



Obrázek 2.3: *Nákres modulu RFM12BP (převzato z [19])*

2.2.1 Základní charakteristiky modulu

Důležité rádiové a elektrické parametry ukazuje následující tabulka:

Tabulka 2.1*:* *Rádiové a elektrické parametry modulu RFM12BP-433*

Kmitočtové pásmo	433 MHz
Modulace	FSK
Přenosová rychlost	max. 115,2 kbps (s interním digitálním filtrem) max. 256 kbps (s externím RC filtrem)
Výstupní výkon	max. 500 mW (nastavitelný v 8 krocích)
Citlivost	- 115 dBm (při BER = 10E-3; Brx = 134 kHz; BR = 1,2 kbps)
Šířka pásma přijímače	67 až 400 kHz (nastavitelná v 6 krocích)
Kmitočtový zdvih modulátoru	15 až 240 kHz (s krokem 15 kHz)
Impedance antény	50 Ω
Napájecí napětí	logické obvody: 2,2 až 3,8 V zesilovač: 12V
Proudový odběr "stand-by"	max. 1,2 mA
Proudový odběr v režimu příjmu	20 až 25 mA
Proudový odběr v režimu vysílání	185 až 200 mA

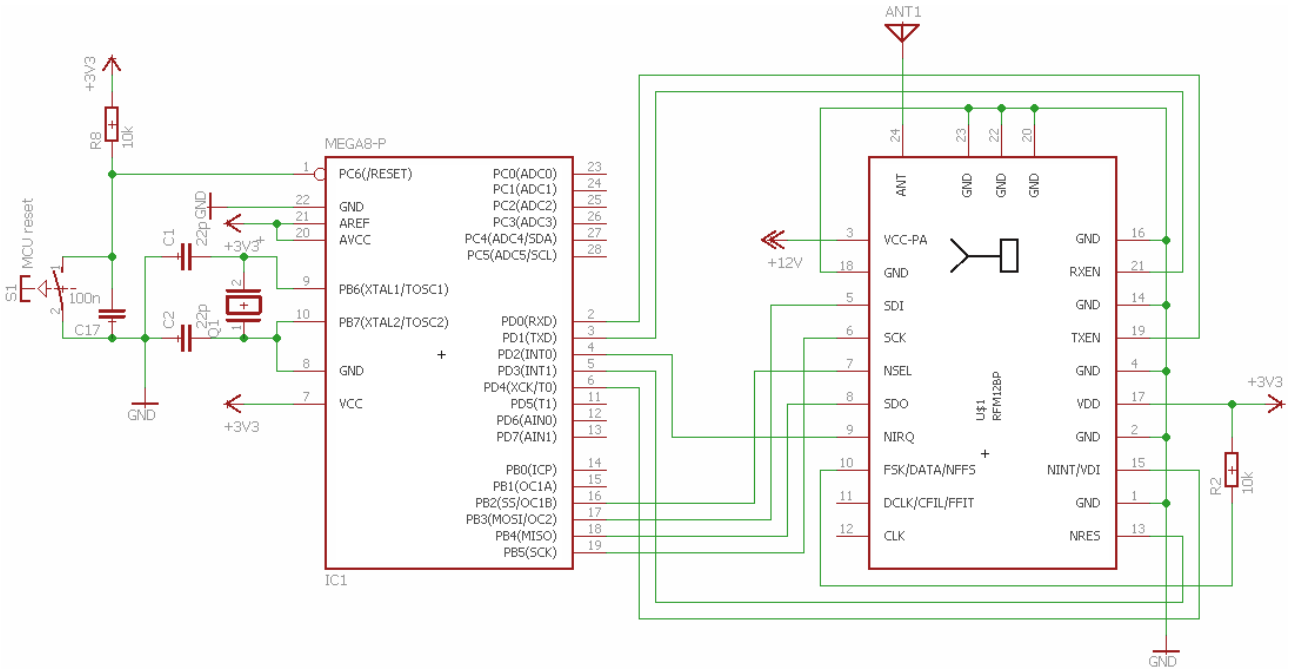
2.2.2 Vývody modulu a jeho připojení k MCU

Rádiový modul je osazen vývody, které slouží pro:

- Napájení: VDD, VCC-PA, GND,
- SPI komunikaci s MCU: SDO, SDI, SCK, nSEL
- Řízení vysílače, přijímače: TXEN, RXEN
- Signalizaci přerušení od modulu: nIRQ
- Indikaci příjmu: nINT/VDI
- Reset modulu: nRES

Na následujícím schématu je ukázáno propojení rádiového modulu s MCU. Pro SPI komunikaci využijeme 4 vodiče. Signál SDO propojíme se vstupem MISO, signál SDI s výstupem MOSI. Dále propojíme vývody hodinového signálu SCK a nakonec linku řídící komunikaci nSEL s vývodem SS. Výhodné (ale nepovinné) je také zapojit signál přerušení od modulu nIRQ na vstup externího přerušení INT0, dále signály pro hardwarové ovládání přijímače (RXEN) a vysílače (TXEN). Nakonec ještě propojíme s MCU signál pro reset modulu nRES a indikaci příjmu dat VDI. Celkem musíme v tomto případě počítat s 9 I/O linkami MCU, přičemž jedna z nich by měla podporovat externí přerušení. Pokud budeme pracovat s vysílacími a přijímacími FIFO registry, což je standardní případ, tak nesmíme zapomenout připojit pin FSK/DATA/NFFS přes 10 kΩ pull-up rezistor na napájecí napětí 3,3 V.

Pokud by se v takto realizovaném zapojení objevovaly určité problémy s linkou nIRQ (neočekávané chování), je možné na ni přidat ještě pull-up rezistor o hodnotě 4,7 kΩ proti napájecímu napětí 3,3 V. Díky tomu zaručíme, že linka bude vždy v jasně definovaném stavu log. 0 nebo log. 1.



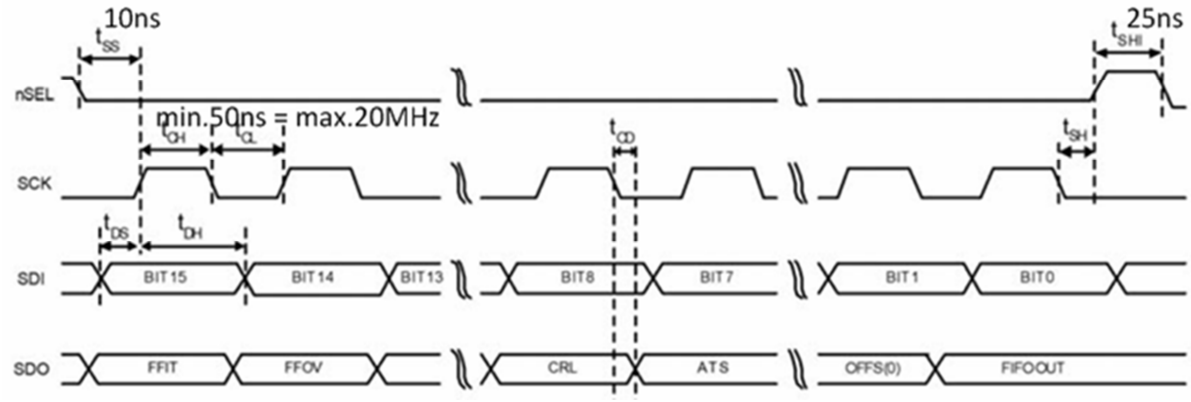
Obrázek 2.4: *Schéma propojení modulu RFM12BP s MCU*

2.2.3 Komunikace SPI

Použitá implementace SPI se nijak neodlišuje od standardního provedení. Komunikace je vždy zahájena ze strany MASTER (MCU), a to nastavením linky Slave Select (nSEL) do úrovně log. 0. Poté je započato vystavování dat současně oběma směry spolu s taktem hodinového signálu. Data jsou odesílána vždy od MSB po LSB v 16 bitovém formátu, kdy první bajt označuje kód instrukce a druhý bajt konkrétní parametry. Po přenosu instrukce je nutné minimálně na 25 ns nastavit linku nSEL do log. 1 pro potvrzení platnosti. MCU nesmí během přenosu jedné instrukce nastavit linku nSEL do log. 1 (typicky po přenosu 1. bajtu), to by zneplatnilo celou instrukci, takže před započetím práce se musíme přesvědčit, že zvolený MCU podporuje přenos více bajtových slov přes rozhraní SPI, případně si napsat vlastní funkci, která takový přenos dat umožní.

U zvolené architektury Atmel AVR lze s výhodou použít vestavěný modul pro SPI rozhraní včetně zdrojového kódu, který je uveden v datasheetu [13] s drobnou úpravou pro možnost přenosu dvou bajtových instrukcí.

Z diagramu ukazujícího časování SPI komunikace mezi MCU a rádiovým modulem vyplývá, že můžeme použít maximální taktovací kmitočet 20 MHz. V datasheetu radiového modulu [20] je ale zároveň uvedena podmínka, že čtení přijímacího FIFO registru nesmí být taktováno rychleji než 2,5 MHz. V neposlední řadě se musíme při návrhu tištěného spoje zamýšlené radiostanice vypořádat s konstrukcí SPI sběrnice jako takové, a to především z hlediska vysokofrekvenčního rušení. Obecným pravidlem je, že signálové cesty by měly být pro dosažení dobré kvality co nejkratší a nejpřímější. Pokud vyjdeme z faktu, že ISP programování MCU Atmel AVR je vlastně SPI přenos, tak nám pro ověření vlastností navržené sběrnice dobře poslouží programátor s měnitelnou rychlostí taktu a následnou verifikací zapsaných dat. Obě tyto podmínky splňuje například již dříve uvedený programátor Presto od firmy Asix spolu s dodávaným programem "AsixUP!". Další variantou ověření vlastností pak může být použití logického analyzátoru s podporou softwarového dekódovaní komunikace (například oblíbené levné klony 8 kanálových analyzátorů Saleae).



Obrázek 2.5: *Komunikace SPI s časováním (převzato z [20])*

2.2.4 Instrukční soubor

V tabulce je uveden přehled jednotlivých instrukcí pro nastavení modulu a následnou rádiovou komunikaci. Jednotlivé instrukce jsou podrobně popsány v příloze A.

Tabulka 2.2: *Instrukce pro obsluhu RFM12BP*

Instrukce	Význam	Hodnota
Configuration Setting	kmitočtové pásmo, kapacita krystalu, povolení práce s Tx a Rx registry	0x80
Power Management	řízení napájení, vysílače a přijímače	0x82
Frequency Setting	nosný kmitočet v rámci pásma	0xA
Data rate	přenosová rychlost	0xC6
Receiver Control	konfigurace VDI, šířky pásma a citlivosti přijímače	0x9
Data Filter	synchronizace datového toku, volba typu filtru, režim obnovení kmitočtu	0xC2
FIFO and Reset Mode	režim plnění Rx FIFO registru, rádiová synchronizace, režim resetu	0xCA
Synchron Pattern	2. synchronizační bajt	0xCE
Receiver FIFO Read	načítání Rx registru FIFO při příjmu dat	0xB0
AFC	automatické řízení kmitočtu	0xC4
Tx Configuration Control	kmitočtový posun FSK modulace, výstupní výkon	0x9
PLL Setting	řízení integrovaného krystalu a výstupu CLK signálu pro MCU, řízení PLL smyčky	0xCC
Transmitter Register Write	zápis do Tx registru při vysílání dat	0xB8
Wake-up Timmer	časovač přechodu modulu do úsporného režimu	0xE0
Low Duty-Cycle	konfigurace snížené spotřeby v režimu přijmu	0xC8
Low Battery Detector and MCU CLK Divider	detektor nízkého napájecího napětí, nastavení kmitočtu na vývod CLK	0xC0
STATUS Read	čtení STATUS registru modulu	0x00

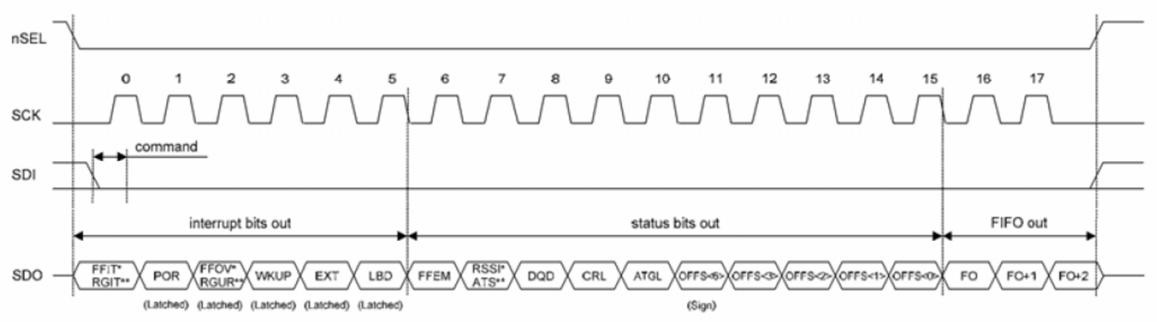
2.2.5 Systém přerušení nIRQ

Jak již bylo uvedeno výše, je výhodné (ale nepovinné) zapojit vývod modulu nIRQ na vstup MCU, který podporuje externí přerušení. Díky tomu může MCU flexibilně reagovat na požadavky modulu při příjmu a vysílání. Přerušení je aktivní, když je nIRQ v log. 0. Následující události aktivují přerušení:

- Tx registr je připraven přijmout další bajt. Díky tomu se nemusíme starat o přesné časování odesílání dat při zvolené přenosové rychlosti. Modul se o přesné časování stará sám.
- FIFO Rx registr přijal nastavený počet bitů. MCU může vykonávat jiné operace a obsloužit modul až v okamžiku, kdy ohlásí, že má nová příchozí data.
- Po zapnutí napájení přečteme status registr a výsledek musí být 0x4000. To značí, že je modul a SPI komunikace v pořádku.
- Wake-up časovač doběhl. V momentě, kdy dojde k probuzení modulu, tak je linka nIRQ nastavena do log. 0 a umožňuje probudit obslužný MCU.
- Napájecí napětí kleslo pod nastavený práh (konfigurační instrukce: Low Battery Detector and MCU CLK Divider).
- FIFO Rx registr není čten dostatečně rychle a přetekl.
- Tx registr není plněn dostatečně rychle, nebo naopak jsou data dodávána příliš rychle a registr přetekl.

2.2.6 **STATUS registr**

Podobně jako u spousty jiných periférií obsahuje i rádiový modul RFM12 stavový registr (STATUS), který můžeme podle potřeby načíst, zjistit aktuální stav zařízení a adekvátně zareagovat. Čtení provádíme instrukcí 0x0000. Jedná se o jedinou instrukci začínající nulou a díky tomu modul pozná, že má odesílat obsah STATUS registru. Časový diagram je ukázán na následujícím obrázku, včetně popisu významu jednotlivých bitů.



Obrázek 2.6: *Časový diagram načítání STATUS registru (převzato z [20])*

Tabulka 2.3: *Význam jednotlivých bitů STATUS registru*

Bit	**Tx**	**Rx**	**Význam**
15 (MSB)	RGIT	FFIT	RGIT = 1 (Tx registr je připraven přijmout další byte) FFIT = 1 (Počet přijatých bitů v Rx FIFO reg. dosáhl nastavené úrovně)
14	POR		POR = 1 (Power on reset)
13	RGUR	FFOV	RGUR = 1 (Data nejsou do Tx reg. dodávána dostatečně rychle, nebo došlo k jejich přepsání) FFOV = 1 (Rx FIFO registr přetekl - data nejsou čtena dostatečně rychle)
12	WKUP		WKUP = 1 (Wake-up časovač doběhl a došlo k probuzení modulu)
11	EXT		EXT = 1 (Na vstupu nINT došlo k žádosti o přerušení ze strany MCU)
10	LBD		LBD = 1 (Napájecí napětí je nižší, než nastavený limit)
9	FFEM		FFEM = 1 (Rx FIFO registr je prázdný)
8	ATS	RSSI	ATS = 1 (detekce silného signálu, nebo nepřizpůsobené antény) RSSI = 1 (síla přijímaného signálu je větší než nastavený limit)
7	DQD		DQD = 1 (přijímaná data odpovídají nastavené požadované kvalitě DQD)
6	CRL		CRL = 1 (Obnovení kmitočtu v pořádku)
5	ATGL		ATGL = 1
4	OFSS6		kmitočtový offset
3	OFSS3		
2	OFSS2		
1	OFSS1		
0 (LSB)	OFFS0		

2.3 Konstrukce hardwaru radiostanice

Před návrhem vlastní radiostanice proběhlo ověření praktické realizovatelnosti popsaného konceptu. Prvním úkolem bylo otestování bloku napájení, především z hlediska dodávaných výkonů, tepelné ztráty stabilizátorů a stability výstupního napětí. Druhým úkolem pak bylo realizovat a odladit zapojení radiového modulu vůči řídícímu MCU. Zde bohužel narazíme na problém originální dokumentace a to poněkud nepřehledné schéma zapojení, kde navíc není ani zakreslen správný typ radiového modulu. Nezbývá tak, než postupně zapojení ladit a sledovat, jak se radiový modul chová.

V této fázi se dobře uplatní nepájivé pole větších rozměrů, kde je možné postavit hned dvě kompletní radiostanice a zároveň tak zkoušet různé způsoby spojení mezi nimi. Během práce ale můžeme v zásadě narazit na dva problémy. Některá nepájivá pole mají poněkud nekvalitně provedené kontakty a občas se stane, že ačkoliv jsou součástky v kontaktech správně zasunuty, není zajištěno spolehlivé elektrické propojení. Druhým problémem představuje samotné propojení součástek pomocí drátů, kdy zjistíme, hlavně u SPI sběrnice, že použití kmitočtů nad 500 kHz začíná být problematické.

2.3.1 Adaptér radiového modulu RFM12BP

Jak již bylo uvedeno dříve, radiový modul RFM12BP je dodáván na DPS, která nemá rozteče vývodů pro standardní THT montáž. Z tohoto důvodu byl nejprve vyroben adaptér, který zajistí, že mezi jednotlivými kolíky v řadě je rozteč 2,54 mm a mezi řadami je pak 27,94 mm. Takto zhotovený adaptér bez problémů pasuje do nepájivého pole. Návrh DPS adaptéru je uveden v příloze B. Vzhledem k tomu, že samotný modul je pájen na adaptér ze strany spojů, tak byl před samotným pájením ze spodní strany potřen lakem za účelem elektrické izolace.

2.3.2 Schéma zapojení

Schéma zapojení radiostanice je uvedeno v příloze C a vychází z doporučených zapojení uvedených v katalogových listech, jednotlivých komponent doplněné o poznatky získané během první fáze ověřování na nepájivém poli.

2.3.3 Návrh desky plošných spojů

K návrhu byl použit oblíbený software Eagle v edici Light. Ta obsahuje omezení týkající se maximálního rozměru výsledné DPS na 100 x 80 mm. Toto omezení prakticky přineslo nutnost rozdělit návrh na dva dílčí celky a to blok napájení a vlastní radiostanici (rádiový modul, řídící MCU a převodník). Pro každý celek byla navržena zvláštní DPS a následně byly oba návrhy spojeny do jednoho výsledku pomocí grafického programu. Součástí grafického post procesu byla také optimalizace izolačních mezer mezi jednotlivými signálovými cestami a zvětšení některých pájecích bodů, pro snadnější výrobu. Nakonec ještě došlo na úpravu zemnícího polygonu tak, aby zabíral co největší plochu. Výsledný layout DPS je uveden v příloze D. Má sice poněkud větší rozměry, než je nutné, ale za to jde velmi dobře vyrobit metodou pozitivní foto cesty na jednostrannou cuprexitovou desku.

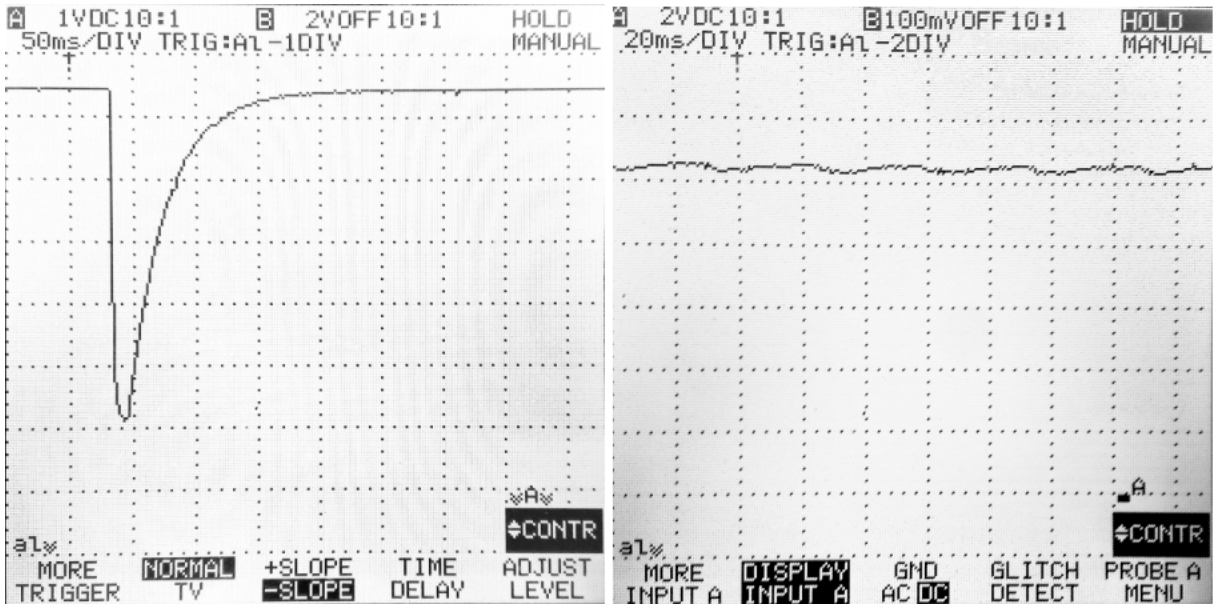
Další výhodou výsledného návrhu je, že díky delším vzdálenostem mezi blokem napájení, řídícím MCU a radiovým modulem nedochází k nechtěnému ovlivňování těchto komponent vlivem elektromagnetického vyzařování a tím pádem není nutné vytvářet stínící kryty.

2.3.4 Výroba a oživení radiostanice

Výroba vlastní radiostanice neskýtala žádné komplikace. Osazovací plán je uveden v příloze D a seznam součástek v příloze E. Po sestavení všech komponent je nutné ještě přidat 6 kusů drátových propojek:

- Připojení převodníku IC2 na napájecí napětí 3,3 V a GND.
- Připojení napájecího napětí 12 V k pinu číslo 2 patice JP1.
- Připojení rezistoru R1 k pinům číslo 8,9 patice JP2.
- Propojení pinu PB4 (Slave Select) na straně MCU s pinem nSEL na straně RFM12BP.
- Propojení pinu PD2 (EXT_INT0) na straně MCU s pinem nIRQ na straně RFM12BP.

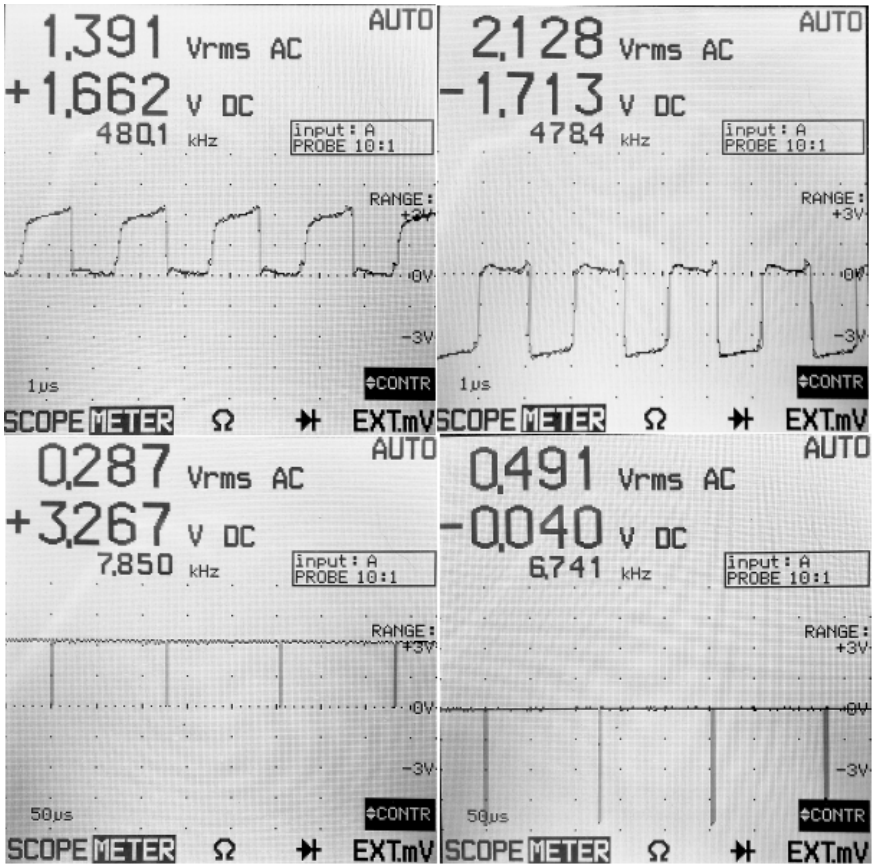
Radiostanice po zapnutí napájení okamžitě naběhla, poté proběhlo nastavení výstupního napětí stabilizátoru LM217T na maximální hodnotu. Během prvotních zkoušek se zjistilo, že stanice má relativně krátký dosah při značném nárůstu chybovosti. Měřením bylo zjištěno značné kolísání napájecího napětí výkonového VF zesilovače a to až o 5,5V. Podrobným zkoumáním byla objevena chyba v zapojení stabilizátoru LM217T. Po správném zapojení se kolísání napětí výrazně zlepšilo na cca. 0,5 V. Průběhy napětí před a po opravě ukazují následující průběhy z osciloskopu.



Obrázek 2.7: *Průběh napájecího napětí VF zesilovače před a po úpravě zapojení*

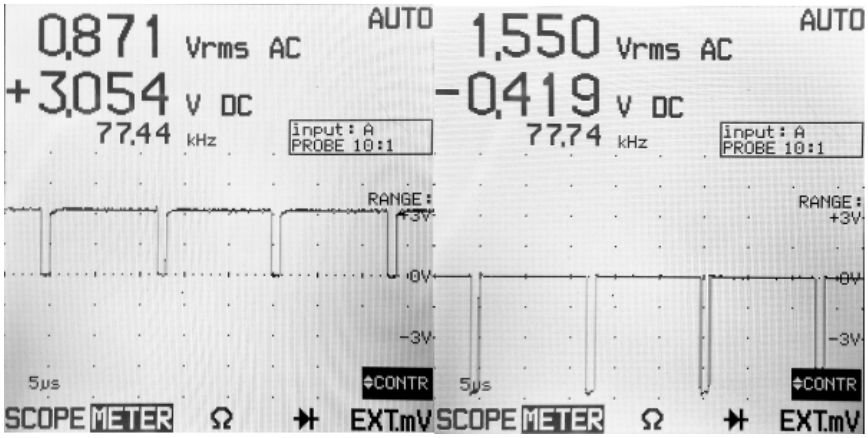
U druhé radiostanice byl experimentálně použit v roli stabilizátoru obvod 78S12. Zvlnění výstupního napětí bylo téměř totožné, a to i při přímém propojení výstupu stabilizátoru s napájecím pinem rádiového modulu pomocí izolovaného měděného vodiče s průřezem 1 mm.

Během fáze vývoje softwaru byly obě stanice připojeny ke stolnímu počítači pomocí sériových rozhraní implementovaných na interní PCI kartě a nevyskytl se žádný problém s komunikací. Potíže nastaly u jedné ze stanic v okamžiku, kdy byly obě připojeny k přenosnému počítači pomocí USB/RS232 převodníku typu HL-340. Proměřením osciloskopem obou převodníku MAX3232CPE byly zjištěny zásadní rozdíly v průběhu napětí na pinech číslo 1,3,4 a 5, které náleží kondenzátorům C12 a C13 v roli nábojových pump integrovaného zdroje převodníku.



Obrázek 2.8: *Průběhy napětí na pinech 3 a 5 převodníků MAX3232CPE při použití USB/RS232 převodníku HL-340*

Horní dva průběhy byly změřeny na převodníku, který pracoval správně, spodní dva pak na problémovém převodníku. Rozdíl je zcela patrný ve spínacím kmitočtu nábojových pump, kdy v prvním případě se pohybuje okolo hranice 480 kHz a v druhém mezi 6 až 8 kHz. Problém by mohl ležet v nekvalitních kondenzátorech C12 a C13. Ještě před jejich výměnou jsme ale zkusili použít jiný typ převodníku, tentokrát s čipem Oxford 952 v podobě Express karty od firmy I-Tec. Po připojení problémy okamžitě zmizely. Průběhy napětí na pinech 3 a 5 ukazuje následující obrázek.



Obrázek 2.9: *Průběhy napětí na pinech 3 a 5 převodníku MAX3232CPE při použití RS232 Express Card I-Tec*

Z průběhů je patrné, že došlo k částečnému rozkmitání nábojových pump. Ačkoliv se výsledná hodnota kmitočtu 77,74 kHz ani zdaleka nepřibližuje požadovaným 480 kHz, zřejmě je to již hodnota dostačující k zajištění správného napájecího napětí.

Ze zvědavosti ještě došlo na měření napěťových úrovní všech dostupných RS232 čipů/převodníků s poměrně očekávaným výsledkem. Nejlépe si vedl RS232 integrovaný na základní desce MSI P43T-C51 stolního počítače. V klidovém stavu měly linky napětí +/- 10,92 V. Další v pořadí byla PCI karta Axago PCEA-PS s napětím +/- 6 V. Následovala již zmiňovaná Express karta I-Tec s napětím +/- 5,67 V. Nejhůře dopadl USB/RS232 převodník HL-340, který nejen že měl napěťové úrovně 0 V, ale navíc indukoval síťový brum 50 Hz o velikosti 0,639 V RMS (i přes to, že byl připojen v přenosném počítači, který nebyl napájen síťovým zdrojem). Průběhy napětí linek jednotlivých převodníků jsou umístěny v příloze F.

Závěr tedy zní, že se zde protnuly dva problémy současně. Za prvé, nekvalitní součástková základna a za druhé, nekvalitní USB/RS232 převodník. Na základě získaných zkušeností můžeme prohlásit, že minimálně během fáze vývoje hardwaru a ladění zapojení je nutné se opřít alespoň o kvalitní převodník, který nám omezí vznik podobných "záhadných" chyb. V momentě kdy jsem si jisti, že máme spolehlivě pracující zařízení, pak můžeme zkusit použít i méně kvalitní převodník, který je ale téměř 10x levnější.

2.4 Softwarové řešení radiostanice

Software je napsán v programovacím jazyku C, který v sobě kombinuje vlastnosti objektového přístupu a zároveň je dostatečně nízko úrovňový pro potřeby 8 bit MCU. Nejnáročnější na celém řešení bylo zjistit správný postup inicializace radiového modulu a naprogramovat funkce pro obsluhu přijímače a vysílače s ohledem na systém přerušení nIRQ.

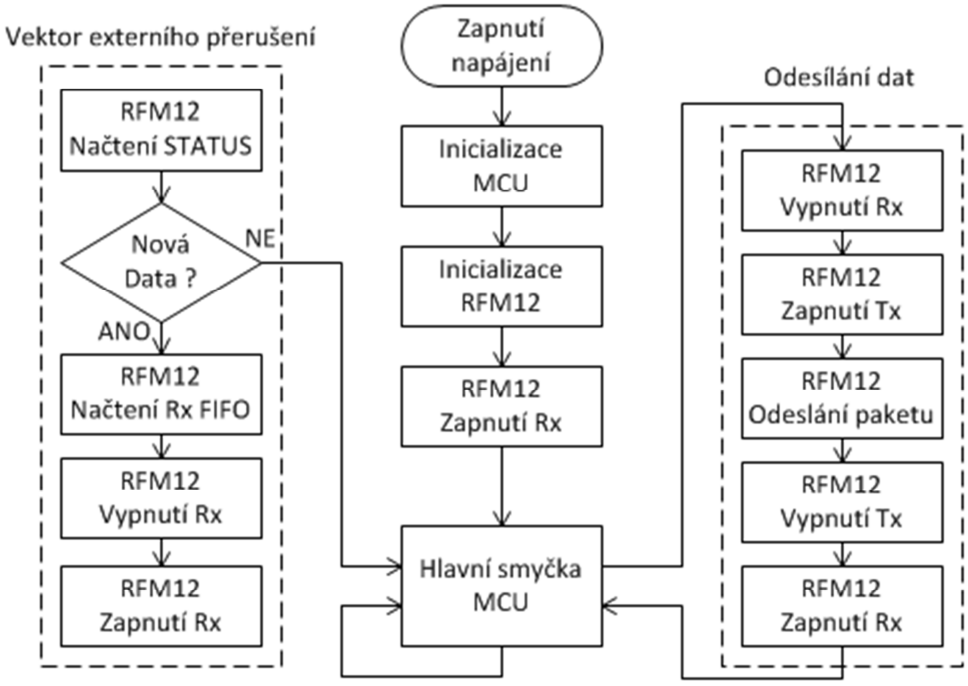
Mezi další výhody zvoleného programovacího jazyka patří možnost rozdělit projekt do jednotlivých zdrojových souborů a ty mezi sebou propojit tzv. hlavičkovým souborem. To výrazně zpřehledňuje práci, kdy jednotlivé funkční celky mohou být vzájemně odděleny. A nakonec nesmíme opomenout možnost softwarového ladění (debug) pomocí programu AVR Studio, jehož vydavatelem je sama společnost Atmel, a který umožňuje odlaďovat zdrojové kódy vytvořené externími IDE.

2.4.1 Koncept řídícího programu

Návrh softwaru se opírá o dvě základní myšlenky:

- Přijímač nikdy dopředu nemůže vědět, kdy bude mít příchozí data.
- Pokud už jsou zachycena nová příchozí data, má obsluha přijímače nejvyšší prioritu, aby nedošlo k přetečení přijímacího FIFO registru (mimo speciální případ ARQ, kde vysílač vždy čeká na kladnou odpověď).

Pro praktickou realizaci to znamená, že pokud nebude radiostanice vysílat, tak zbytek času bude v režimu příjmu a dále musíme využít systému externího přerušení MCU, který zajistí potřebnou okamžitost obsluhy radiového modulu.



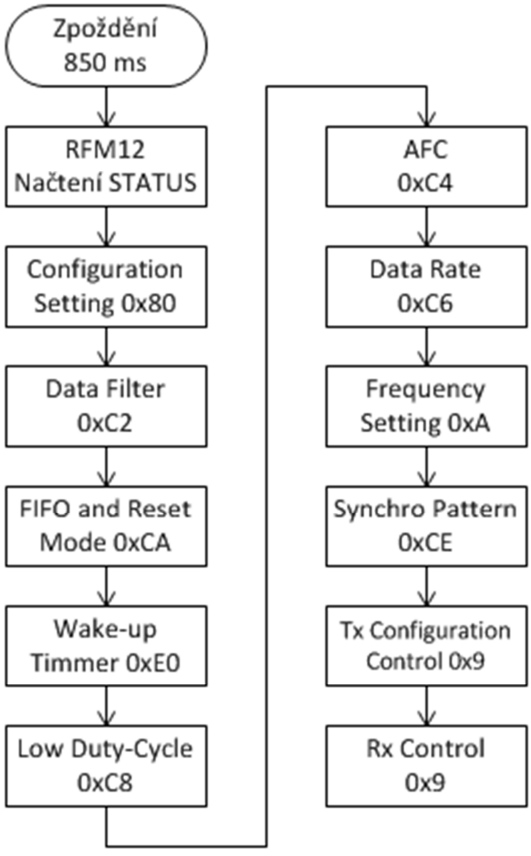
Obrázek 2.10: *Koncept řídícího programu radiostanice*

2.4.2 Inicializace radiového modulu RFM12BP-433

Po zapnutí napájení 3,3 V se řídící obvod radiového modulu uvede do stavu POR. To znamená, že se do všech registrů, sloužících k nastavení radiového modulu, zapíší výchozí hodnoty, které jsou uvedeny v datasheetu. V tuto chvíli v podstatě můžeme začít okamžitě s modulem pracovat, tj. vysílat a přijímat data. Ve většině aplikací ale chceme změnit některé parametry. Obvykle se jedná alespoň o nosný kmitočet, vysílací výkon a přenosovou rychlost. Tyto změny můžeme udělat poměrně snadno zápisem příslušných instrukcí a odpovídajících parametrů. Dobrým pomocníkem je konfigurační kalkulátor dostupný na internetu [24]. V našem konkrétním případě měníme tyto parametry:

- nosný kmitočet,
- šířka pásma přijímače,
- citlivost přijímače a útlum,
- kmitočtový zdvih vysílače,
- výstupní výkon vysílače,
- synchronizační bajt.

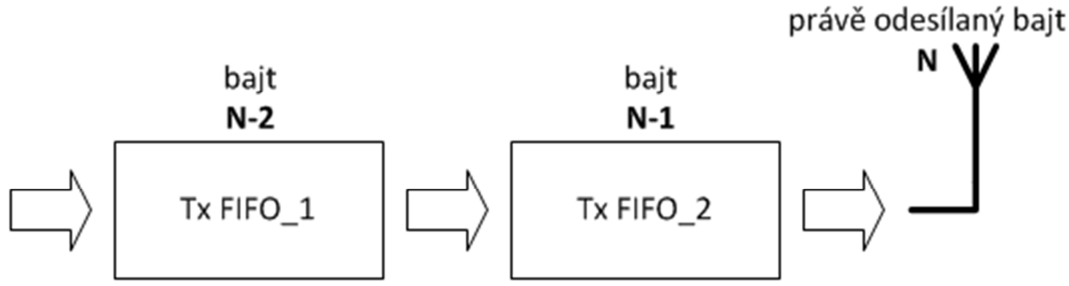
Veškeré změny od výchozího nastavení uděláme hned při startu radiostanice ještě před započetím radiového provozu. Pořadí instrukcí není nikde stanoveno, takže je možné volit prakticky libovolnou kombinaci. S výhodou využíváme u řídícího MCU paměť EEPROM, pro zaznamenání aktuálních parametrů a jejich uchování i při vypnutí napájení.



Obrázek 2.11: *Vývojový diagram inicializace radiového modulu RFM12BP*

2.4.3 **Režim vysílání**

Odesílání dat se děje postupným zápisem do vysílacího FIFO registru. Důležité je dodržet správné časování, aby nedošlo k podtečení. Za tímto účelem vhodně využijeme vlastností linky nIRQ, která přechodem ze stavu log. 1 do stavu log. 0 (aktivní přerušení) signalizuje, že je radiový modul připraven přijmout další bajt. Struktura vysílacích registrů je ukázána na následujícím obrázku.

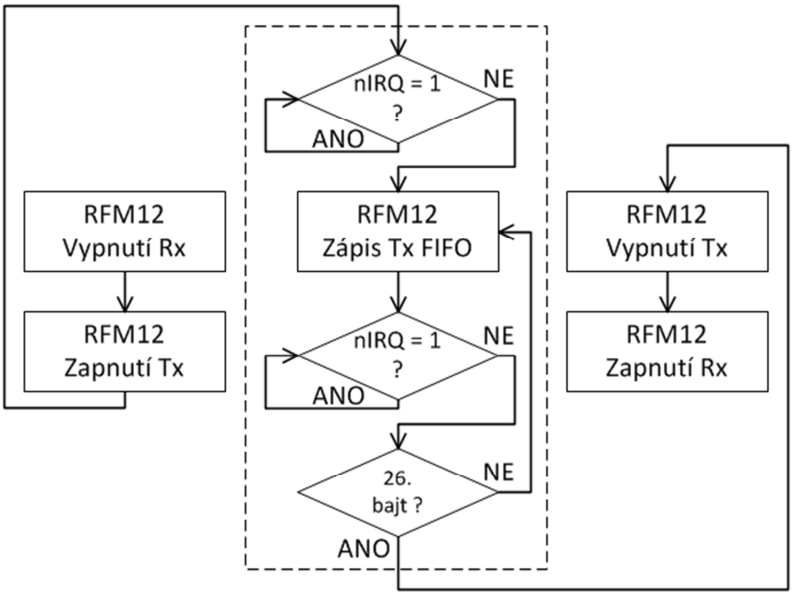


Obrázek 2.12: *Blokové schéma vysílacího FIFO registru*

Z obrázku je patrné, že pokud zapíšeme instrukcí "*Transmitter Register Write*" první bajt do radiového modulu, nedojde k jeho okamžitému odvysílání. K tomu je potřeba odeslat ještě další dva bajty, než se první postupně posune přes druhý FIFO registr, až do vysílače. Podobný problém je pak na konci paketu, kdy po posledním bajtu, který chceme odvysílat, musíme do modulu zapsat ještě dva bajty navíc (ty však zůstanu ve FIFO registrech a nebudou odvysílány).

Radiový modul RFM12BP je vybaven vlastním systémem synchronizace a identifikace. To znamená, že těsně před vlastním odesláním datového paketu musíme odvysílat dvojici bajtů o hodnotě 0xAA (což binárně představuje 0b10101010), díky které se přijímač synchronizuje s vysílačem. Dále pak musíme odvysílat druhou dvojici identifikačních bajtů, z nichž první je povinný a má vždy hodnotu 0x2D, druhý je nepovinný, uživatelsky nastavitelný instrukcí "*Synchron Pattern*" a defaultně má hodnotu 0xD4. Přijímač můžeme nastavit tak, že aktivuje přerušení nIRQ až v okamžiku, kdy je korektně synchronizován a zároveň zachytí dvojici identifikačních bajtů. Praktické pokusy ale ukázaly, že přijímač spouští přerušení velmi často, v závislosti na šířce pásma, až 10x za minutu a to i přes to, že vysílač právě nepracoval. Proto byl do datového paketu navíc přidán ještě jeden synchronizační bajt o hodnotě 0xCA (viz. kapitola 1.5.2). Následující obrázek ukazuje celkovou strukturu rádiového rámce včetně synchronizace, identifikace a ukončující dvojice bajtů.

Obrázek 2.13: *Celková struktura rádiového rámce*



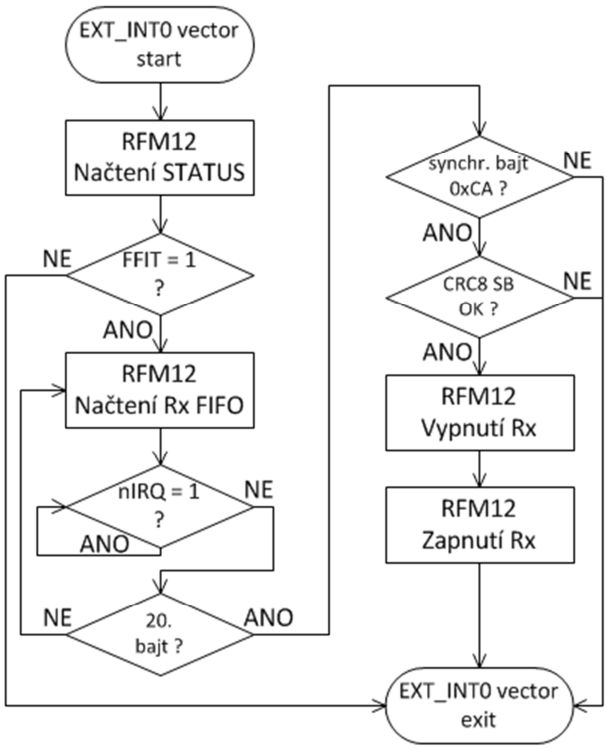
Obrázek 2.14: *Vývojový diagram funkce pro odeslání dat*

Z vývojového diagramu je patrné, jak odesílání dat funguje. Nejprve musíme postupně vypnout přijímač a zapnout vysílač pomocí instrukce "*Power Management*". Následně sledujeme stav linky nIRQ a čekáme, až ji rádiový modul nastaví do log. 0, což je pro nás signálem, že modul očekává další data. Postupně odesíláme celý radiový rámec po jednotlivých bajtech pomocí instrukce "*Transmitter Register Write*" a po každém bajtu znovu sledujeme linku nIRQ. Po odeslání posledního bajtu vypneme vysílač a zapneme přijímač, opět pomocí instrukce "*Power Management*".

2.4.4 **Režim příjmu**

Vzhledem k tomu, že přijímač nikdy dopředu nezná okamžik příchodu nových dat, je opět využita linka nIRQ, která je v MCU připojena k pinu PD2, jenž slouží zároveň jako vstup externího přerušení EXT_INT0. Pokud přijímač zachytil synchronizační bajty a následně i korektní identifikaci, tak nastaví linku nIRQ do log. 0. Pro MCU je to signál, že má okamžitě spustit vektor externího přerušení. V něm nejprve načteme STATUS registr pro ověření, jestli je nastaven bit FFIT. Pokud ano, pokračuje funkce dále a postupně pomocí instrukce "*Receiver FIFO Read*" načítá jednotlivé nově příchozí bajty, a to vždy, když je linka nIRQ nastavena modulem do log. 0. Díky tomu, že používáme pakety stejné fixní délky, tak je i počet načítaných bajtů roven délce datového paketu, tj. 20. Po posledním načteném bajtu je přijímač vypnut a znovu zapnut, díky čemuž modul vždy nastaví linku nIRQ zpět do log. 1.

Vektor přerušení dále zkontroluje hodnotu synchronizačního bajtu (0xCA) a kontrolní součet servisního bajtu. Pokud je některá z kontrol vyhodnocena jako neúspěšná, je celý paket zahozen a vektor ukončen, v opačném případě je nastaven příznak v globálním registru, že došlo ke korektnímu příjmu dat a paket je předán k dalšímu zpracování podle informací v servisním bajtu.



Obrázek 2.15: *Vývojový diagram funkce pro příjem dat*

2.4.5 **Softwarové řízení toku sériové linky RS232**

Režim radiostanice "SERIAL MODEM" je, jak už název napovídá, koncipován jako bezdrátové prodloužení sériové linky osobního počítače. Přenos dat je řešen tak, že v RAM paměti řídícího MCU je vytvořen buffer o celkové velikosti 860 B, který je postupně plněn novými příchozími daty z linky RS232. Po naplnění bufferu se data následně skládají do paketů, podle nastavení zabezpečení přenosu se provede výpočet FEC, případně šifrování AES a pakety se odesílají přijímači. V případě, že chceme odeslat soubor o velikosti přesahující 860 B, je nutné realizovat řízení toku sériové linky, protože řídící MCU má omezené systémové prostředky a nedokázal by příchozí data zpracovat a odeslat v reálném čase.

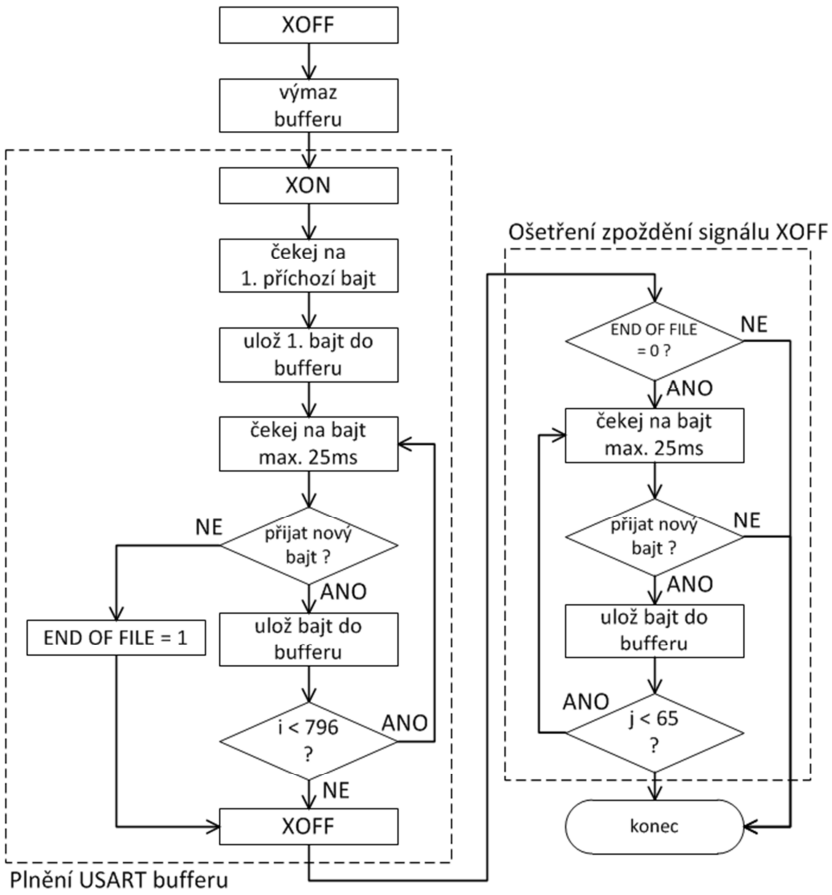
Řízení toku sériové linky rozeznáváme dvojího druhu:

- Hardwarové (DTR/DSR nebo RTS/CTS),
- Softwarové (XON/XOFF).

Hardwarové řízení je v podstatě efektivnější, jelikož přijímač může přesně regulovat okamžik, kdy se má přísun nových dat zastavit. Určitou nevýhodou je nutnost použití dalších dvou signálových linek. To nepředstavuje problém, pokud máme počítač, který je na základní desce vybaven sériovým rozhraním, případně máme přídavnou kartu, která obsahuje odpovídající řadič. V dnešní době ale drtivá většina počítačů sériové rozhraní již nemá, a proto je nutné využívat různé USB/RS232 převodníky. Jejich cenové rozpětí a kvalita je velmi široká a u levných modelů je potencionální riziko, že nebudou mít implementovány přídavné linky pro řízení toku.

Softwarového řízení toku naopak nevyžaduje žádné další linky navíc, díky tomu bude možné využít prakticky jakýkoliv převodník. Toto řešení má ale jednu velkou nevýhodu, která vyplývá ze samotné podstaty věci. Pokud se nám buffer v MCU zaplní a následně pošle signál XOFF pro zastavení přísunu nových dat, tak trvá určitou dobu, než tento signál projde z nejnižší hardwarové úrovně, přes operační systém počítače až k aplikaci, která následně zastaví odesílání dat. Po tuto dobu jsou data neustále odesílána, a pokud není tento stav ošetřen, dojde k přeplnění MCU bufferu.

Řešení problému spočívá v určení předstihu s jakým se má signál XOFF odeslat ještě před zaplněním bufferu tak, aby byl řádně uložen i vzniklý datový "překmit". V našem případě je nastaveno pro rezervu posledních 64 B bufferu. Princip řízení je ukázán na následujícím vývojovém diagramu. Funkce postupně načítá jednotlivé bajty tak, že čeká maximálně 25 ms na nová data. Pokud přijdou, uloží bajt do bufferu a zvýší hodnotu počítadla (proměnná *i*). V momentě kdy počítadlo překročí hodnotu 796 a v bufferu nám zbývá posledních 64 B, odešle signál XOFF a následně začne ukládat data, která ještě přichází vlivem zpoždění signálu. Pro přesné určení kolik bajtů bylo zapsáno v podobě datového "překmitu" slouží druhé počítadlo (proměnná *j*). Celkovou velikost dat v bufferu pak určíme jako součet hodnot proměnných *i* a *j*.



Obrázek 2.16: *Vývojový diagram SW řízení RS232 (XON/XOFF)*

2.5 Testování radiostanic

V souladu se zadáním proběhlo testování vyrobených radiostanic v terénu, které mělo za cíl prověřit jejich chování přímo v reálném prostředí.

Cílem první skupiny měření bylo ověřit dosah rádiového spojení radiostanic pro různé kombinace nastavení, a zároveň otestovat funkčnost algoritmů zabezpečení proti chybám (FEC a ARQ). Druhá skupina měření probíhala v rámci jednoho pracoviště s cílem vyzkoušet funkčnost režimu SERIAL MODEM pro přenos dat, především pak softwarové řízení toku sériové linky XON/XOFF. Dále se pak ověřila reálná doba přenosu testovacího souboru o velikosti 20 KiB při různých komunikačních rychlostech. Třetí skupina se zaměřila na VF část, tj. výkon na anténě a spektrum vyzařovaného signálu. Poslední skupina testování se pak zabývala ověřením některých algoritmů, a to buď za použití logického analyzátoru, anebo pomocí softwarového emulátoru architektury AVR (debugger AVR Studio). Zde se jednalo hlavně o ověření šifrovací funkce AES a testu Hammingova kódu.

2.5.1 Měření dosahu rádiového spojení

Dosah, jakožto vzdálenost, na kterou mají radiostanice ještě spolehlivé spojení, je pravděpodobně nejdůležitější parametr, při návrhu komunikačního řetězce. V praxi si musíme nejdříve definovat jaké informace a na jakou vzdálenost chceme přenášet a až následně potom vybírat konkrétní technické řešení, které nejlépe vyhoví požadavkům. V našem případě jsme sice neměli konkrétní specifikace toho, jakou vzdálenost musí radiostanice překlenout, nicméně ověření tohoto parametru nám pomůže lépe zhodnotit kvalitu výsledného návrhu.

2.5.1.1 Metodika měření

Metodika měření spočívala ve vytvoření dvou pracovišť. První bylo stacionární a skládalo se z přenosného počítače a radiostanice, která byla umístěna 1 metr nad zemí. Druhé pracoviště bylo naopak mobilní, vybavené radiostanicí, přenosným počítačem a přijímačem družicové navigace pro určení přesné zeměpisné polohy.

Volba stacionárního stanoviště byla ovlivněna těmito požadavky:

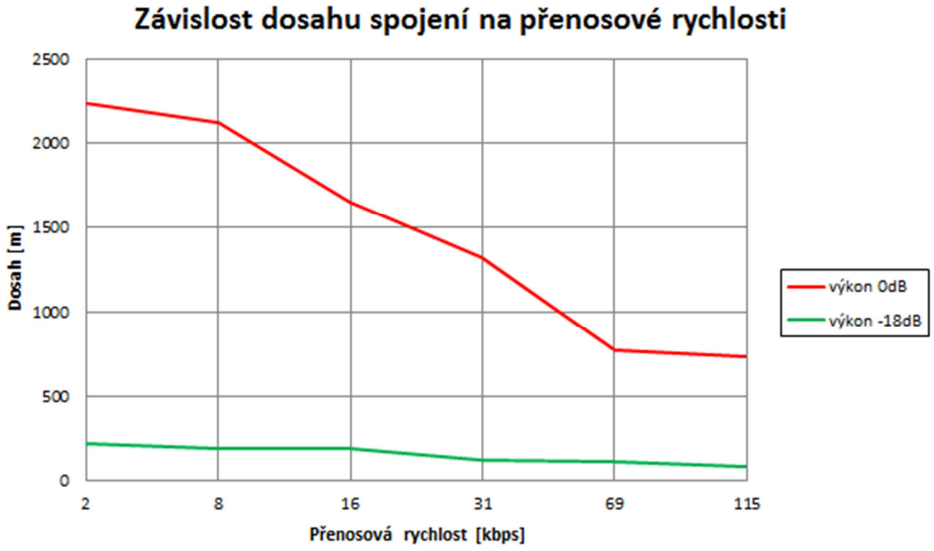
- vyloučení vlivu rušení okolními vysílači pracujícími v ISM pásmu 433 MHz.
- Zajištění přímé viditelnosti mezi pracovišti pro určení maximálního možného dosahu bez ovlivnění zástavbou či krajinou.

Měření dosahu probíhalo na dvou úrovních vysílacího výkonu a to 27 dBm (maximální výkon 500 mW) a 9 dBm (maximální dovolený výkon v ČR dle [7], tj. hodnota 7,81 mW). Na obou výkonových úrovních se dále měnily přenosové rychlosti v rozmezí 2 až 115 kbps v 6tipřed programovaných krocích, které odpovídají zvolené přenosové rychlosti RS232.Samotná realizace probíhala vzdalováním mobilního pracoviště za neustálé kontroly spojení pomocí funkce PING až do okamžiku, kdy docházelo k 10% ztrátě paketů. Tento bod byl označen za místo spolehlivého spojení. Následně proběhl test proti-chybových algoritmů FEC a ARQ pomocí funkce TEXT, kdy si pracoviště mezi sebou domluvila nové parametry přenosové rychlosti a celé měření se zopakovalo.

2.5.1.2 *Naměřené hodnoty*

Tabulka 2.4: *Naměřené hodnoty dosahu rádiového spojení*

přenosová rychlost		výkon	dosah	poznámka
RS232	**rádiová**			
[Bd]	[kbps]	[dBm]	[m]	
1200	1,959	27	2234	Rx BW = 67 kHz freq. dev. = 45 kHz LNA = -14 dB
		9	208	
4800	7,663	27	2120	
		9	182	
9600	15,674	27	1653	
		9	179	
19200	31,348	27	1318	Rx BW = 134 kHz freq. dev. = 60 kHz
		9	113	
38400	68,966	27	775	Rx BW = 200 kHz freq. dev. = 90 kHz
		9	110	
57600	114,943	27	734	Rx BW = 270 kHz freq. dev. = 135 kHz
		9	79	



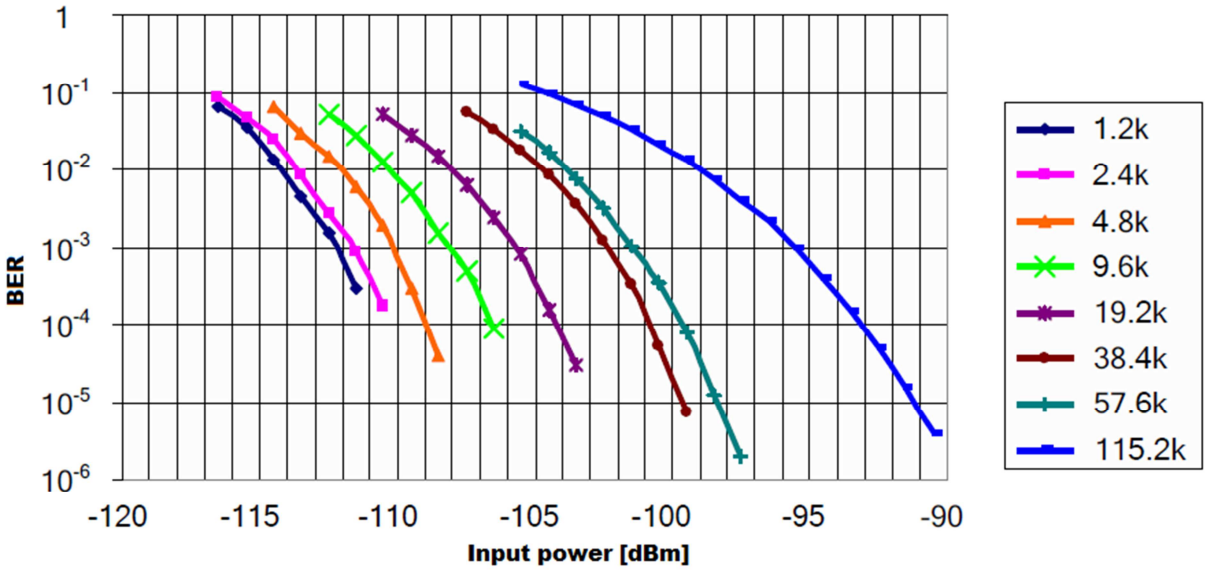
Obrázek 2.17: *Závislost dosahu spojení na přenosové rychlosti*

2.5.1.3 *Zhodnocení*

Z naměřených hodnot vidíme, že dosah spojení je limitován jednak zvoleným vysílacím výkonem, a následně pak i přenosovou rychlostí. Tento poznatek vychází ze vztahu (1.2) a faktu, že výkon šumu roste s šířkou pásma a kvůli tomu klesá při daném výkonu na přijímací straně poměr signál-šum (a tím roste BER až nad použitelnou mez cca. 10E-4). Pro zlepšení dosahu při dodržení nařízení ČTÚ o maximálním dovoleném výkonu tak můžeme provést v zásadě tyto kroky:

- zmenšit šířku pásma, snížením přenosové rychlosti (díky tomu zlepšíme SNR a následně BER),
- použít antény s vyšším ziskem (tím dostaneme na přijímač vyšší výkon a opět zlepšíme SNR a BER).

Závislost BER na velikosti přijímaného signálu je uvedena v [20] na straně 37 a pro pásmo 433 MHz vypadá následovně.



Obrázek 2.18: *Závislost BER na úrovni přijímaného signálu (převzato z [20])*

Z celkového hlediska si radiostanice vedly velmi dobře. Dosah odpovídá předpokladům a plně vyhovuje cílovému zaměření, tj. přenos dat na krátké vzdálenosti. Také algoritmy proti chybového zabezpečení obstály, především pak režim ARQ, který obsluze zaručuje 100% spolehlivost přenosu dat.

2.5.2 Měření přenosu dat a řízení sériové linky XON/XOFF

2.5.2.1 Metodika měření

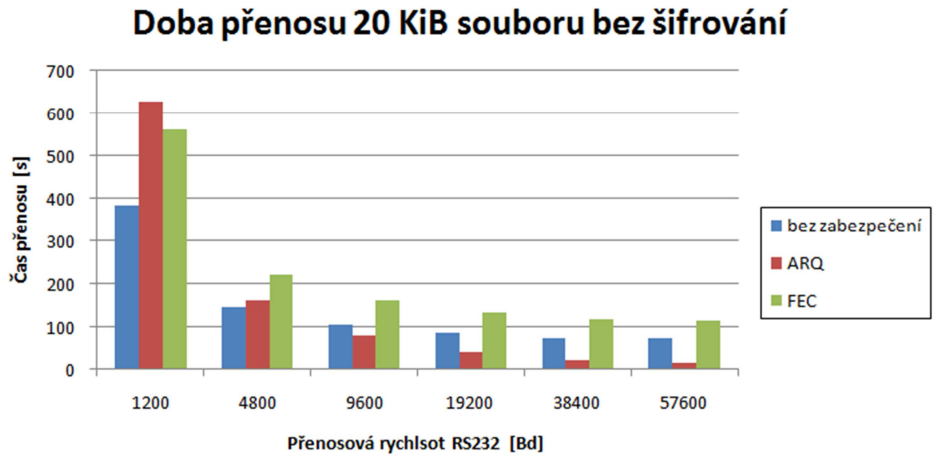
Tentokrát probíhalo měření v laboratorních podmínkách na jednom pracovišti. Obě radiostanice byly připojeny k jedinému počítači, jedna přes převodník USB/RS232 a druhá přes Expres kartu. V počítači byl vytvořen testovací soubor o velikosti 20 KiB ve formátu MS Word. Tento typ souboru byl zvolen proto, že se případná chyba přenosu ihned projeví tím, že přijatý soubor nebude možné otevřít, nebo se při otevírání zobrazí chyba o poškození.

Cílem bylo změřit dobu, která je nutná pro přenos souboru na všechny kombinace zabezpečení proti chybám a šifrování na dostupných komunikačních rychlostech. Výkon byl po celou dobu nastaven na fixní hodnotu 6 dBm. Po zjištění časů bylo následně provedeno srovnání jednotlivých metod a výpočet čisté přenosové rychlosti, tj. takové, která respektuje potřebnou režii rádiového spojení a řízení sériové linky.

2.5.2.2 *Naměřené hodnoty*

Tabulka 2.5: *Doba přenosu 20 KiB souboru*

přenosová rychlost		**bez zabezpečení**		**ARQ**		**FEC**		**poznámka**
RS232	**rádiová**	**bez šifrování**	**AES**	**bez šifrování**	**AES**	**bez šifrování**	**AES**	
[Bd]	[kbps]	[s]	[s]	[s]	[s]	[s]	[s]	
1200	1,959	385	399	628	631	565	562	Rx BW = 67 kHz freq. dev. = 45 kHz LNA = -14 dB
4800	7,663	146	160	162	164	221	221	
9600	15,674	105	120	81	82	163	164	
19200	31,348	86	100	42	43	135	135	Rx BW = 134 kHz freq. dev. = 60 kHz
38400	68,966	76	90	21	24	120	120	Rx BW = 200 kHz freq. dev. = 90 kHz
57600	114,943	75	88	15	17	115	115	Rx BW = 270 kHz freq. dev. = 135 kHz

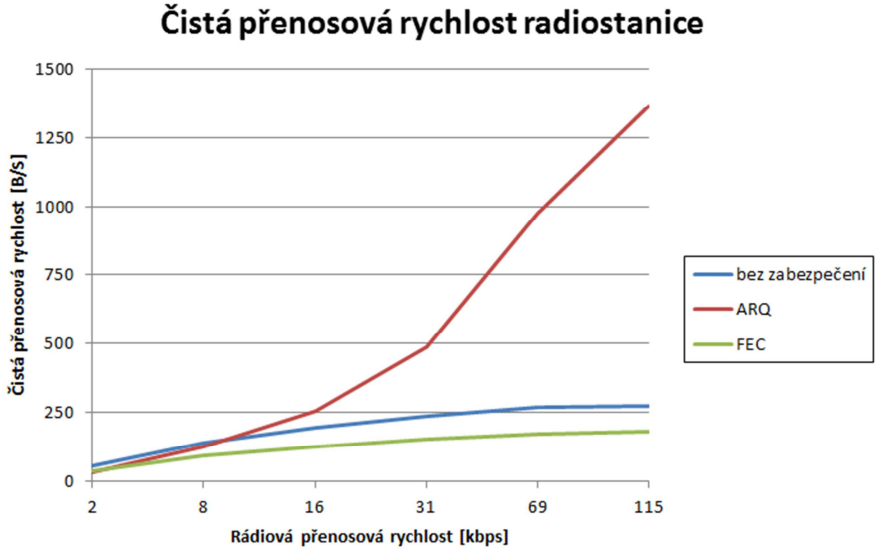


Obrázek 2.19: *Doba přenosu 20 KiB souboru bez šifrování AES*

2.5.2.3 *Zhodnocení*

Měření ukázalo, že použití šifrovacího algoritmu AES prakticky nezmění potřebnou dobu pro přenos. Můžeme tak říct, že algoritmus je na 8 bitovém MCU Atmel AVR skutečně velmi výkonný. Nejlépe si vedl režim zabezpečení ARQ, a to proto, že je každý paket ihned po svém zpracování přijímačem potvrzen a vysílač může pokračovat. U ostatních režimů je nastavena fixní doba čekání řádově v desítkách ms. Postupným testováním a optimalizací by bylo možné tyto doby zkrátit až na nejnutnější minimum, a tím výrazně zvýšit výslednou rychlost. Pro ARQ na rychlosti 1200 Bd ale můžeme vidět, že je nejpomalejší. Je to způsobeno tím, že nejprve je naplněn buffer vysílače, data jsou odvysílána příjemci, a ten následně buffer vyprázdní a až poté posílá vysílači signál, že je připraven. To je pro něj pokyn k novému plnění vysílacího bufferu. Algoritmus by bylo možné vylepšit tak, aby se vyprazdňování bufferu na straně přijímače a plnění na straně vysílače dělo současně, tím by se potřebná doba ještě výrazněji zkrátila.

Na základě naměřených hodnot byla dále vypočítána čistá přenosová rychlost. Graficky znázorněná na následujícím obrázku.



Obrázek 2.20: *Čistá přenosová rychlost radiostanice*

I v tomto testu radiostanice obstála, i když se ukázaly určité nedostatky v některých režimech přenosu. Řízení sériové linky naopak fungovalo naprosto bezchybně. Program RealTerm pro operační systém Windows ukazoval, kolik bajtů bylo v každém datovém proudu posláno a vždy se jednalo o rozmezí 796 až 860 B. Z toho vyplývá, že koncept zachytávání datového "překmitu" po odeslání signálu XOFF se osvědčil.

2.5.3 Měření rádiového spektra a výstupního výkonu VF zesilovače

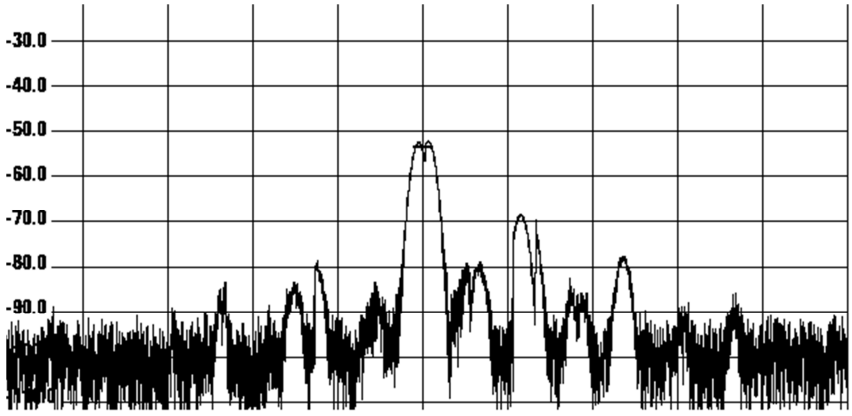
2.5.3.1 Metodika měření

Měření proběhlo v laboratoři EB209 pomocí spektrálního analyzátoru a VF watt metru firmy Rohde & Schwarz. Radiostanice byly umístěny přibližně 3 metry od sebe a následně byl zahájen přenos souboru v režimu ARQ na přenosové rychlosti 9600 Bd, čemuž odpovídá rádiová přenosová rychlost 15,674 kbps. Výkon byl nastaven na úroveň 9 dBm (7,81 mW). Spektrální analyzátor vybaven vlastní anténou, monitoroval rádiové spektrum v oblasti ISM pásma 433 MHz.

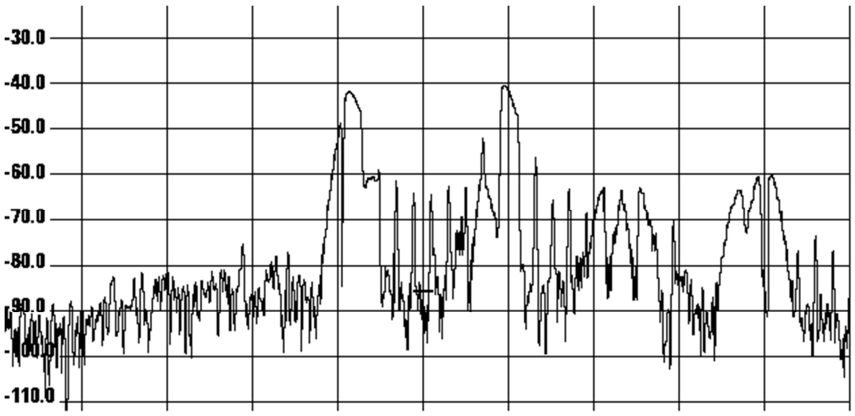
Výstupní výkon byl změřen pomocí VF wattmetru připojeného přes koaxiální kabel o impedanci 50 Ω přímo na výstup zesilovače. Pro omezení rušivých vlivů byl na radiostanici nastaven maximální výkon 27 dBm (500 mW).

2.5.3.2 Naměřené hodnoty

Výstupní výkon VF zesilovače při nastavené úrovni 27 dBm (500 mW) dosáhl naměřené hodnoty 510 mW.



Obrázek 2.21: *Rádiové spektrum RFM12BP-433 (span: 8 MHz)*



Obrázek 2.22: *Rádiové spektrum RFM12BP-433 (span: 500 kHz)*

2.5.3.3 ***Zhodnocení***

Z hlediska výstupního výkonu i rádiového spektra moduly RFM12BP-433 splnily výrobcem deklarované parametry.

2.5.4 **Ověření funkce algoritmů AES a Hammingova kódu**

2.5.4.1 ***Metodika měření***

Vzhledem k tomu, že nebyla k dispozici třetí radiostanice, která by odposlouchávala provoz, bylo využito logického analyzátoru Saleae, který monitoroval aktivitu na SPI rozhraní jedné z radiostanic. Nejprve byl připraven zkušební text "*Hello Word! 123*", který byl v podobě textové zprávy odeslán proti-stanici a pomocí analyzátoru se ověřila správnost odeslaných bajtů do rádiového modulu. Následně bylo zapnuto šifrování a analyzátor opět zachytil bajty na SPI, tentokrát v zašifrované podobě. Zachycená zpráva byla porovnána s výsledkem z on-line AES kalkulátoru [25], tím byla ověřena správnost šifrovacího algoritmu.

Ověření Hammingova kódu (8,4) probíhalo čistě na úrovni softwarového emulátoru vestavěného v AVR Studiu, kde byly postupně simulovány jednonásobná i více násobná chyba.

2.5.4.2 ***Naměřené hodnoty***

Tabulka 2.6: *Ukázka šifrování textu*

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.
Nešifrovaný text																
ASCII	H	e	l	l	o		W	o	r	d	!		1	2	3	
Hex	48	65	6c	6c	6f	20	57	6f	72	64	21	20	31	32	33	00
Šifrovaný text AES 128 bit																
ASCII	♂	↕	■	A	t		E	W	▼	.	d	R	·	ď		h
Hex	0b	12	df	41	74	20	45	57	1f	2e	64	52	fa	d4	20	68
Klíč																
Hex	2b	7e	28	16	3c	4f	f7	22	fa	11	20	01	aa	bb	cc	dd

Ukázka zachycené komunikace na SPI sběrnici je umístěna v Příloze H.

2.5.4.3 ***Zhodnocení***

Z analýzy zachycené komunikace a srovnáním s on-line kalkulátorem AES vyplynulo, že šifrování funguje přesně podle očekávání, navíc, jak se ukázalo při měření datových přenosu, je napsáno i velmi úsporně a ani malá 8 bitová architektura nemá s implementací problém. Ověření Hammingova kódu bylo také úspěšně provedeno, během praktických testů pak přijímač občas hlásil, kolik bajtů bylo obnoveno, případně poškozeno.

Závěr

V teoretické části práce byly nastíněny základní bloky komunikačního řetězce a problémy, se kterými se můžeme setkat, především při bezdrátové komunikaci. Došlo ke shrnutí metod zabezpečení komunikace jak proti chybám, tak proti úmyslnému odposlechu. S ohledem na zvolenou architekturu řídícího mikropočítače byly dále provedeny podrobné rozbory algoritmů pro výpočet rozšířeného Hammingova kódu a detekčního kódu CRC s délkou 8 bitů. Zvláštní pozornost byla věnována šifrování, srovnání výhod a nevýhod jednotlivých typů kryptografie z hlediska distribuce klíčů. Na základě porovnání a s ohledem na určení finálních radiostanic, byl zvolen symetrický šifrovací algoritmu Rijndael, známý jako AES, jehož princip šifrování je schematicky naznačen. Předposlední část teoretického rozboru se zabývá kmitočtovým pásmem ISM. V přehledové tabulce jsou naznačeny jednotlivé úseky v kmitočtovém plánu a případná omezení kladená národní autoritou. Na konec s ohledem na datový přenos, je probrána problematika řízení rádiového přenosu. Jsou zde tabulkově uvedeny jednotlivé režimy spojení a možnosti zabezpečení. V návaznosti na tato fakta je zde proveden návrh vlastního datového paketu, který je určen přímo pro konkrétní aplikaci. Podrobně je popsán význam jednotlivých položek datového paketu, včetně odůvodnění proč bylo zvoleno právě toto konkrétní řešení.

Praktická část se zabývá aplikací teoretických poznatků na reálný výrobek - mikropočítačem řízenou radiostanici v pásmu ISM. V úvodu je naznačeno blokové schéma a koncept radiostanice, následuje podrobný popis jednotlivých bloků s volbou konkrétní součástkové základny. Následuje podrobný popis rádiového modulu RFM12BP-433. Uvedeny jsou hlavní technické parametry, způsob propojení modulu a mikropočítače s popisem komunikačního protokolu SPI, včetně instrukčního souboru. Nechybí popis systému přerušení a STATUS registru. V další části je popsána výroba vlastní radiostanice s uvedením celkového schéma zapojení a návrhu DPS, včetně osazovacího plánu a seznamu součástek. Vzhledem k tomu, že se jednalo o prototypy, ani nám se nevyhnuly problémy s oživením zapojení. Při této příležitosti byl proveden malý test běžně dostupných převodníků pro rozhraní RS232, jakožto jedné z klíčových součástí pro zprovoznění radiostanic. V kapitole věnující se softwarovému řešení jsou podrobně popsány algoritmy pro obsluhu modulu RFM12BP. V části věnované vysílání najdeme návrh celého rádiového rámce, což je původní datový paket doplněný ještě o další bajty na základě informací z datasheetu. Vzhledem k tomu, že se nejedná o triviální problém, je také podrobně popsán algoritmus pro řízení přenosu dat přes sériové rozhraní typu XON/XOFF.

Poslední část je věnována testování a měření hotových radiostanic. Měření se zaměřilo především na dosah rádiového spojení v terénu pro různé přenosové rychlosti a vysílací výkony. Výsledky splnily očekávání, když se podařilo realizovat spojení až na 2234 m. Z hlediska proti-chybových kódů, které byly rovněž testovány, si o něco lépe vedl režim ARQ a to z prostého důvodu, že v případě neúspěšného spojení vypsal obsluze zprávu na terminál. Další měření proběhlo v laboratorních podmínkách a mělo ověřit schopnost radiostanice v roli datového modemu sériové linky RS232. Měřením čisté přenosové rychlosti se ukázala určitá neefektivita algoritmů, které v pár případech způsobovaly zbytečné zdržení a celkovou degradaci výkonových parametrů. Dalším kolem optimalizací by však bylo možné přenosové rychlosti zvýšit. Měření rádiového spektra a výstupního výkonu už žádné překvapení nepřineslo. Modul se choval přesně, jak výrobce deklaruje, jak z hlediska spektra, tak i výstupního výkonu VF zesilovače. Ověření funkce šifrovacího algoritmu AES se muselo omezit pouze na čtení komunikace SPI rozhraní mezi mikropočítačem a modulem. Následné srovnání

zachycených dat s výsledky on-line AES kalkulátoru potvrdily, že algoritmus je plně funkční a navíc v souladu s předchozím měřením i poměrně výkonný, a to i na 8 bitové architektuře. Podobně byl ověřen i algoritmus Hammingova kódu, který musel během simulace jednonásobné chyby provést její detekci a korekci a v případě vícenásobné chyby tuto detekovat. V obou případech obstál.

Vlastním přínosem práce je především souhrnný popis vlastností rádiového modulu RFM12BP, podrobný popis všech jeho součástí, doplněný o vlastní poznatky získané během konstrukce radiostanice. Neméně důležité je pak zpracování funkcí v programovacím jazyce C pro inicializaci, zápis a čtení dat. Funkce jsou v maximální míře zjednodušeny, využívající pouze základních příkazů a knihoven tak, aby odpovídaly normě ANSI C. Jsou bohatě okomentovány pro snadnou použitelnost v budoucích projektech. Dalším důležitým přínosem je ověření konceptu přenosu dat pro radiostanice krátkého dosahu. Zde se jedná hlavně o formát rádiového paketu a význam servisního bajtu, jakožto řídícího prvku, který umožní konstrukci automatizovaného přijímače. Velký význam mají také algoritmy pro zabezpečení rádiového přenosu proti chybám. U režimu ARQ se jedná o výpočet CRC8, který je implementován pomocí look-up tabulky a pro režim FEC jde o implementaci rozšířeného Hammingova kódu (8,4). Obě metody prokázaly dobré vlastnosti a paměťovou nenáročnost na zvolené architektuře.

Z hlediska dalšího vývoje práce je možné provést celkovou optimalizaci zdrojového kódu pro zvýšení přenosové rychlosti, nebo kód upravit pro výrazně výkonnější platformu jako například populární procesory ARM v jednodeskových minipočítačích. Tím by mohlo být dosaženo přenosu dat bez zpoždění v podobě načítání bufferu. Další možností by pak bylo vybavit radiostanici například snímačem pohybu a realizovat bezdrátový přenos poplachových informací elektronického zabezpečení. Spolu s výkonným šifrováním a zabezpečením přenosu dat proti chybám bychom dostali systém, který by vynikal pokročilými funkcemi za vynaložení poměrně malých finančních nákladů.

Použitá literatura

[1] NĚMEC, Karel. *Datová komunikace*. Vyd. 1. Brno: VUTIUM, 2000. ISBN 80-214-1652-1.

[2] HANUS, Stanislav. *Bezdrátové a mobilní komunikace*. Vyd. 1. Brno: Vysoké učení technické, 2001. ISBN 80-214-1833-8.

[3] ŽALUD, Václav. *Moderní radioelektronika*. 1. vyd. Praha: BEN - technická literatura, 2000. ISBN 80-86056-47-3.

[4] DOBEŠ, Josef a Václav ŽALUD. *Moderní radiotechnika*. 1. vyd. Praha: BEN - technická literatura, 2006. ISBN 80-7300-132-2.

[5] *Český telekomunikační úřad* [online]. [cit. 2016-04-10]. Dostupné z: http://www.ctu.cz/

[6] *Český telekomunikační úřad: Využití rádiového spektra* [online]. [cit. 2016-04-10]. Dostupné z: http://spektrum.ctu.cz/

[7] *Všeobecné oprávnění č. VO-R/10/04.2012-7 k využívání rádiových kmitočtů a k provozování zařízení krátkého dosahu*. Praha: Český telekomunikační úřad, 2012.

[8] KOOPMAN, Philip a Tribid CHAKRAVARTY. *Cyclic Redundancy Code (CRC) Polynomial Selection For Embedded Networks*. The International Conference on Dependable Systems and Networks, 2004.

[9] DVORSKÝ, Marek a Pavel NEVLUD. *Přenos dat: učební text*. Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, 2012. ISBN 978-80-248-2604-2.

[10] *Federal Information Processing Standards Publication 197: Specification for the ADVANCED ENCRYPTION STANDARD (AES)*. National Institute of Standards and Technology (NIST), 2001.

[11] Advanced Encryption Standard. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-04-10]. Dostupné z: https://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[12] *AES - šifra pro třetí tisíciletí: Softwarová implementace* [online]. [cit. 2016-04-10]. Dostupné z: http://www.jikos.cz/~gumysh/docs/AES/Pages/Page05.html

[13] *Atmel 8-bit AVR Microcontroller with 32K Bytes In-Systém Programmable Flash: ATmega32A* [online]. [cit. 2016-04-10]. Dostupné z: http://www.atmel.com/images/atmel-8155-8-bit-microcontroller-avr-atmega32a_datasheet.pdf

[14] *LFXX: Very low drop voltage regulator with inhibit function* [online]. [cit. 2016-04-10]. Dostupné z: http://www.st.com/web/en/resource/technical/document/datasheet/CD00000546.pdf

[15] *LM217, LM317: 1.2 V to 37 V adjustable voltage regulators* [online]. [cit. 2016-04-10]. Dostupné z: http://www.st.com/web/en/resource/technical/document/datasheet/CD00000455.pdf

[16] *MAX3222/MAX3232/ MAX3237/MAX3241: 3.0V to 5.5V, Low-Power, up to 1Mbps, True RS-232 Transceivers Using Four 0.1µF External Capacitors* [online]. [cit. 2016-04-10]. Dostupné z: http://pdfserv.maximintegrated.com/en/ds/MAX3222-MAX3241.pdf

[17] MATUSZCZYK, Jacek. Antény prakticky. 1. české vyd. Praha: BEN - technická literatura, 2002. ISBN 80-7300-084-9.

[18] *OK1IKE: ANTÉNNÍ ODKAZY* [online]. [cit. 2016-04-10]. Dostupné z: http://ok1ike.c-a-v.com/hamradio_soubory/ant_odkazy.htm

[19] *UNIVERSAL ISM BAND FSK TRANSCEIVER MODULE WITH 500mW OUTPUT POWER: RFM12BP* [online]. [cit. 2016-04-10]. Dostupné z: http://www.hoperf.com/upload/rf/RFM12BP.PDF

[20] *RFM12B Universal ISM Band FSK Transceiver* [online]. [cit. 2016-04-10]. Dostupné z: http://www.hoperf.com/upload/rf/RFM12B.pdf

[21] *Mikrocontroller.net durchsuchen: RFM12* [online]. [cit. 2016-04-10]. Dostupné z: http://www.mikrocontroller.net/articles/RFM12

[22] *Gobotronics's Blog: RFM12 programming* [online]. [cit. 2016-04-10]. Dostupné z: https://gobotronics.wordpress.com/2010/10/07/rfm12-programming/

[23] *Strobotics: RFM12 Tutorial* [online]. [cit. 2016-04-10]. Dostupné z: http://blog.strobotics.com.au/2008/01/08/rfm12-tutorial-part1/

[24] *Web Tools - JeeLabs: RFM12B Command Calculator* [online]. [cit. 2016-04-10]. Dostupné z: http://tools.jeelabs.org/rfm12b.html

[25] *AES Calculator* [online]. [cit. 2016-04-16]. Dostupné z: http://testprotect.com/appendix/AEScalc

Seznam příloh

Příloha na CD/DVD.

Adresářová struktura přiloženého CD/DVD:

- návrh DPS
  - adapter RFM12BP
  - radiostanice
- přeložený hex soubor
- zdrojové kódy

Příloha A: *Instrukční soubor RFM12BP*

Tabulka A.1: *Configuration Setting Command 0x80*

el	ef	b1	b0	x3	x2	x1	x0

Popis:

- Povoluje přístupu k interním datovým Tx registrům (v tom případě musí být pin FSK připojen na log. 1) (*el*)
- Povoluje práci s Rx FIFO registrem (registr pro příjem dat) (*ef*)
- Nastavuje kmitočtové pásmo, ve kterém modul pracuje (433, 868, 915 MHz) (*b1 - b0*)
- Nastavuje kapacitu integrovaného krystalu (v rozmezí 8,5 - 16 pF) (*x3 - x0*)

Výchozí nastavení registru (POR = 0x08):

- Přístup k datovým registrům zakázán
- Kmitočtové pásmo nenastaveno
- Kapacita integrovaného krystalu 12,5 pF

Tabulka A.2: *Power Management Command 0x82*

er	ebb	et	es	ex	eb	ew	Dc

Popis:

- Zapíná celý přijímací řetězec (*er*)
- Zapíná pásmovou propusť(jednu z částí příjmového řetězce) (*ebb*)
- Zapíná PLL smyčku, výkonový zesilovač a zapíná vysílání (*et*)
- Zapíná syntetizátor (*es*)
- Zapíná integrovaný krystal (oscilátor) (*ex*)
- Zapíná detekci nízkého stavu baterie (*eb*)
- Zapíná wake-up timmer (*ew*)
- Vypíná výstup integrovaného krystalu na výstup CLK (*dc*)

Výchozí nastavení registru (POR = 0x08):

- Vysílač i přijímač vypnuty
- Integrovaný krystal zapnut
- Detekce nízkého napětí napájecí baterie vypnuta
- Wake-up timmer vypnut
- Na výstupu CLK je signál z integrovaného krystalu

Tabulka A.3: *Frequency Setting Command 0xA*

f11	f10	f9	f8	f7	f6	f5	f4	f3	f2	f1	f0

Popis:

- Nastavení nosného kmitočtu radiového modulu podle zvoleného pásma (*f11 - f0*)

Výchozí nastavení (POR = 0x680):

- Pro RFM12BP-433 je kmitočet 430,26 MHz

Tabulka A.4: *Data Rate Command 0xC6*

cs	r6	r5	r4	r3	r2	r1	r0

Popis:

- Nastavení bitové rychlosti rádiového přenosu (*r6 - r0*)
- Nastavení před děliče 1/8 pro jemné nastavení přenosové rychlosti (*cs*)

Výchozí nastavení (POR = 0x23):

- Přenosová rychlost 9,579 kbps

Tabulka A.5: *Receiver Control Command 0x9*

0	p16	d1	d0	i2	i1	i0	g1	g0	r2	r1	r0

Popis:

- Nastavuje funkci pinu nINT/VDI (*p16*)
- Nastavuje odezvu VDI (*d1 - d0*)
- Nastavuje šířku pásma přijímače Rx BW (*i2 - i0*)
- Nastavuje zisk vstupního zesilovače LNA (LowNoiseAmplifier) (*g1 - g0*)
- Nastavuje práh úrovně RSSI (*r2 - r0*)

Výchozí nastavení (POR = 0x080):

- Pin nINT/VDI je nastaven na nINT
- VDI nastaveno na rychlou odezvu
- Šířka pásma 200 kHz
- Zisk LNA nastaven na 0 dB (žádné zeslabení)
- RSSI nastaveno na -103 dBm

Tabulka A.6: *Data Filter Command 0xC2*

al	ml	1	s	1	f2	f1	f0

Popis:

- Nastavuje automatické ovládání clock recovery (*al*)
- Nastavuje režim clock recovery v případě manuálního ovládání - má význam pouze pro digitální filtr (*s=0*)v manuálním režimu (*al = 0*) (*ml*)
- Volba typu filtru (*s*)
- Nastavení DQD prahu (*f2 - f0*)

Výchozí nastavení (POR = 0x2C):

- Manuální režim clock recovery v pomalém módu
- Digitální filtr
- DQD=4

Tabulka A.7: *FIFO and Reset Mode Command 0xCA*

f3	f2	f1	f0	sp	al	ff	Dr

Popis:

- Nastavuje, po kolika datových bitech (mimo synchronizaci) dojde k přerušení - signál nIRQ (*f3 - f0*)
- Volba délky synchronizačního řetězce (*sp*)
- Podmínky, za kterých se začne plnit FIFO (*al*)
- Povolení plnit FIFO (*ff*)
- Nastavení "citlivého" resetu (*dr*)

Výchozí nastavení (POR = 0x80):

- Přerušení po 8 datových bitech
- 2 synchronizační bajty (výchozí hodnota 0x2DD4)
- Plnění FIFO je zakázáno a po povolení bude plněno po synchronizačním řetězci
- Citlivý reset zapnut

Tabulka A.8: *Synchron Pattern Command 0xCE*

b7	b6	b5	b4	b3	b2	b1	b0

Popis:

- Nastavuje druhý bajt identifikace (*b7 - b0*)

Výchozí nastavení (POR = 0xD4):

- Druhý bajt identifikace nastaven na hodnotu 0xD4

Tabulka A.9: *AFC Command 0xC4*

a1	a0	rl1	rl0	st	fi	oe	En

Popis:

- Nastavuje režim měření kmitočtu (*a1 – a0*)
- Nastavuje maximální odchylky pracovního kmitočtu (*rl1 – rl0*)
- Povoluje uložení offsetu do STATUS registru (*st*)
- Nastavuje režim vysoké přesnosti měření (*fi*)
- Povoluje použití offset registru (spodní 4 bity STATUS registru) (*oe*)
- Povoluje výpočet kmitočtového offsetu (odchylky) (*en*)

Výchozí nastavení (POR = 0xF7):

- měření AFC bez ohledu na VDI
- Odchylky nastaveny na nejpřísnější (+3f ... -4f)
- Hodnota kmitočtového offsetu není ukládána do STATUS registru
- Režim vysoké přesnosti
- Výpočet kmitočtového offsetu povolen

Tabulka A.10: *Transmitter Configuration Control Command 0x9*

1	0	0	mp	m3	m2	m1	m0	0	p2	p1	p0

Popis:

- Nastavuje polaritu modulace (*mp*)
- Nastavuje modulační kmitočtový zdvih (*m3 – m0*)
- Nastavuje výstupní výkon zesilovače (*p2 – p0*)

Výchozí nastavení (POR = 0x00):

- Positivní polarita modulace
- Modulační zdvih 15kHz
- Výstupní výkon zesilovače 500 mW

Tabulka A.11: *Low Battery and MCU Clock Divider Command 0xC0*

d2	d1	d0	0	v3	v2	v1	v0

Popis:

- Nastavuje dělič kmitočtu integrovaného krystalu na výstupu CLK pro MCU (*d2 – d0*)
- Nastavuje práh napětí pro detekci nízkého stavu baterie (*v3 – v0*)

Výchozí nastavení (POR = 0x00):

- Na výstup CLK je přiveden kmitočet 1 MHz
- Práh napětí baterie nastaven na 2,2 V

Tabulka A.12: *PLL Setting Command 0xCC*

0	ob1	ob0	lpx	ddy	ddit	1	bw0

Popis:

- Nastavuje strmost nástupné a sestupné hrany signálů pro MCU, pokud je připojen na CLK vývod (*ob1 - ob0*)
- Nastavuje low power režim pro integrovaný krystal (*lpx*)
- Povoluje zpoždění fázového detektoru (*ddy*)
- Povoluje dithering z PLL (*ddit*)
- Nastavuje šířku pásma PLL smyčky (*bw0*)

Výchozí nastavení (POR = 0x76):

- Strmost hran nastavená pro 2,5 MHz
- Doba náběhu integrovaného krystalu 2 ms
- Zpoždění fázového detektoru zakázáno
- Dithering vypnut
- Max. bitrate až 256 kbps

Tabulka A.13: *Wake-up Timmer Command 0xE*

r3	r2	r1	r0	m7	m6	m5	m4	m3	m2	m1	m0

Popis:

- Nastavení periody wake-up časovače (*r3 - m0*)

Výchozí nastavení (POR = 0x196):

- Hodnota R = 1
- Hodnota M = 150
- perioda časovače nastavena na 300 ms

Tabulka A.14: *Low Duty-Cycle Command 0xC8*

d6	d5	d4	d3	d2	d1	d0	En

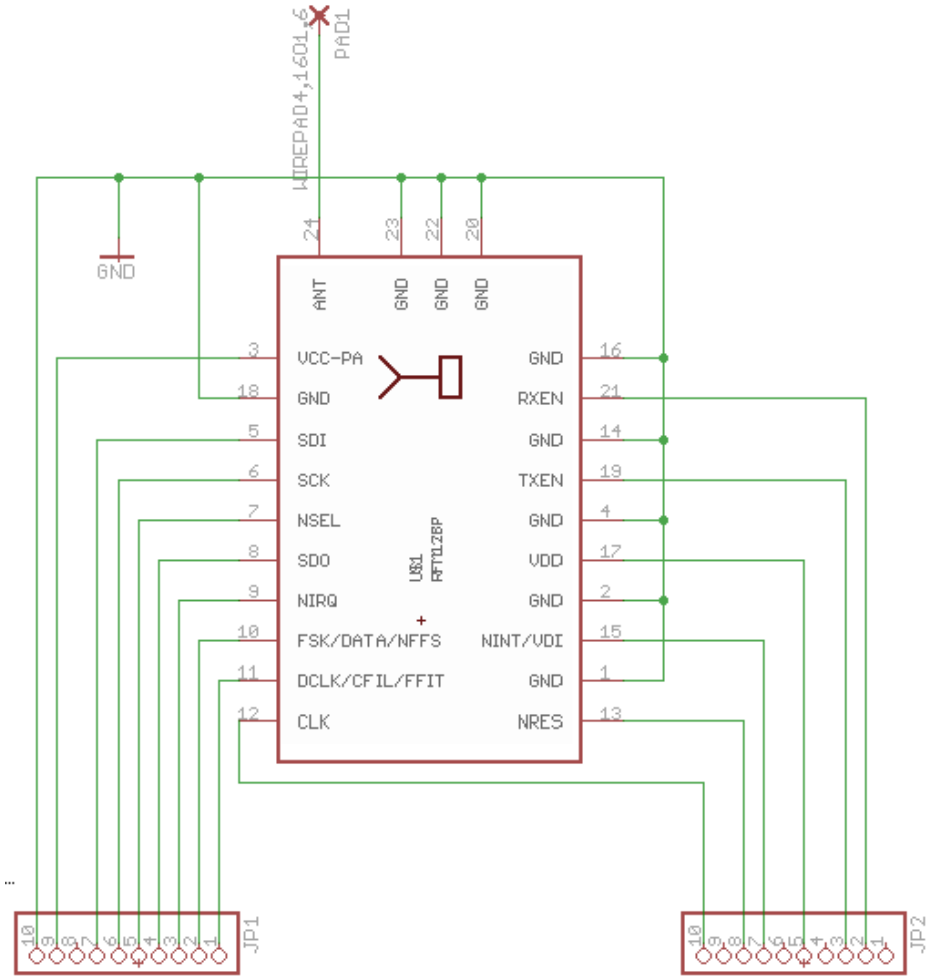
Popis:

- Nastavení Low Duty-Cycle (*d6 - d0*)
- Povoluje režim Low Duty-Cycle Command (*en*)

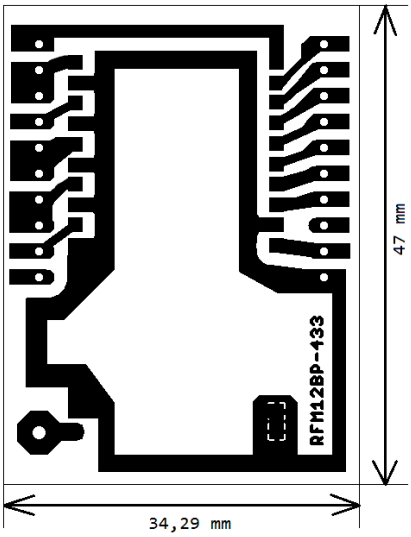
Výchozí nastavení (POR = 0x0E):

- Low Duty-Cycle zakázáno

Příloha B: *Adaptér rádiového modulu RFM12BP*

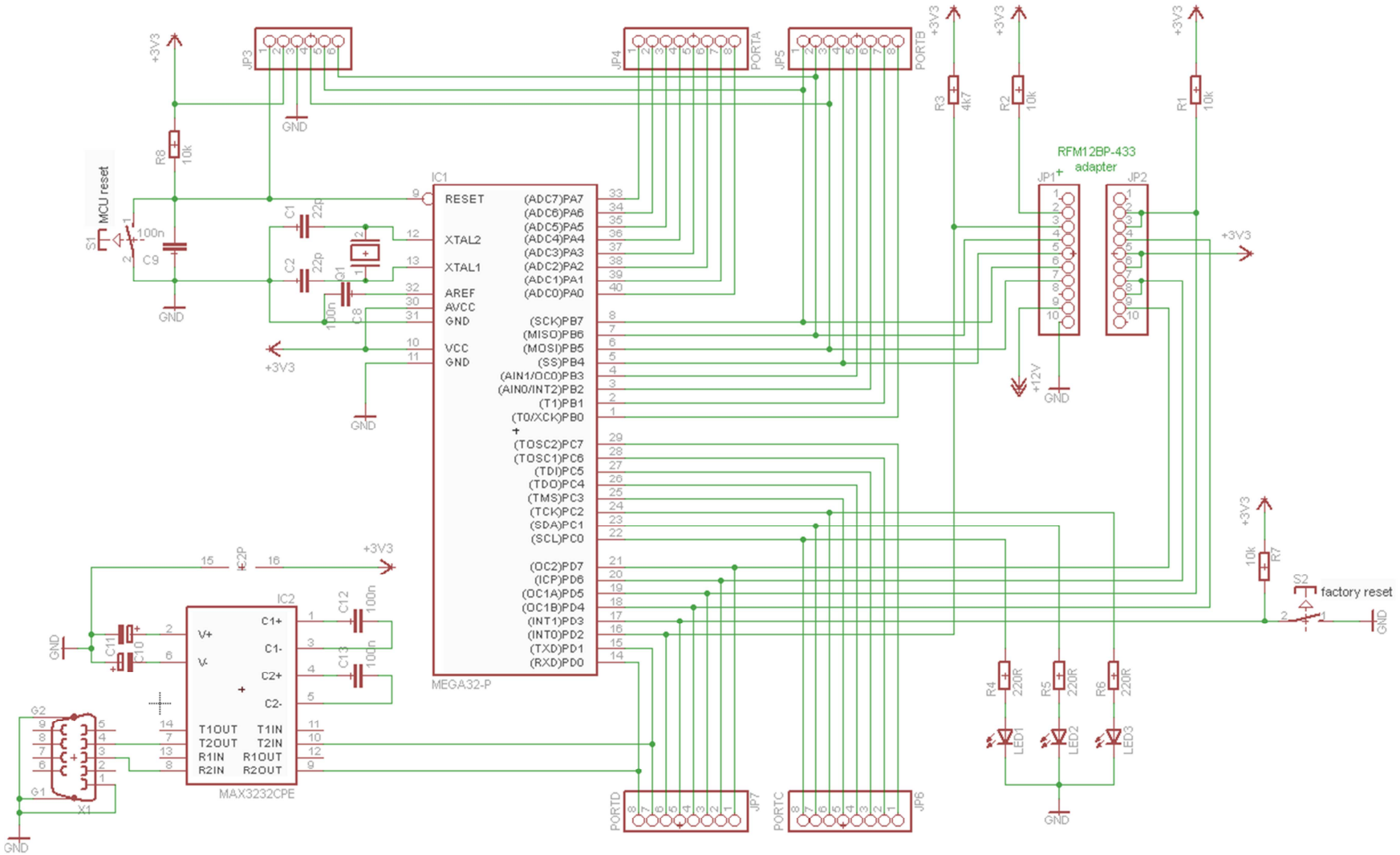


Obrázek B.1: *Schéma zapojení adaptéru RFM12BP*



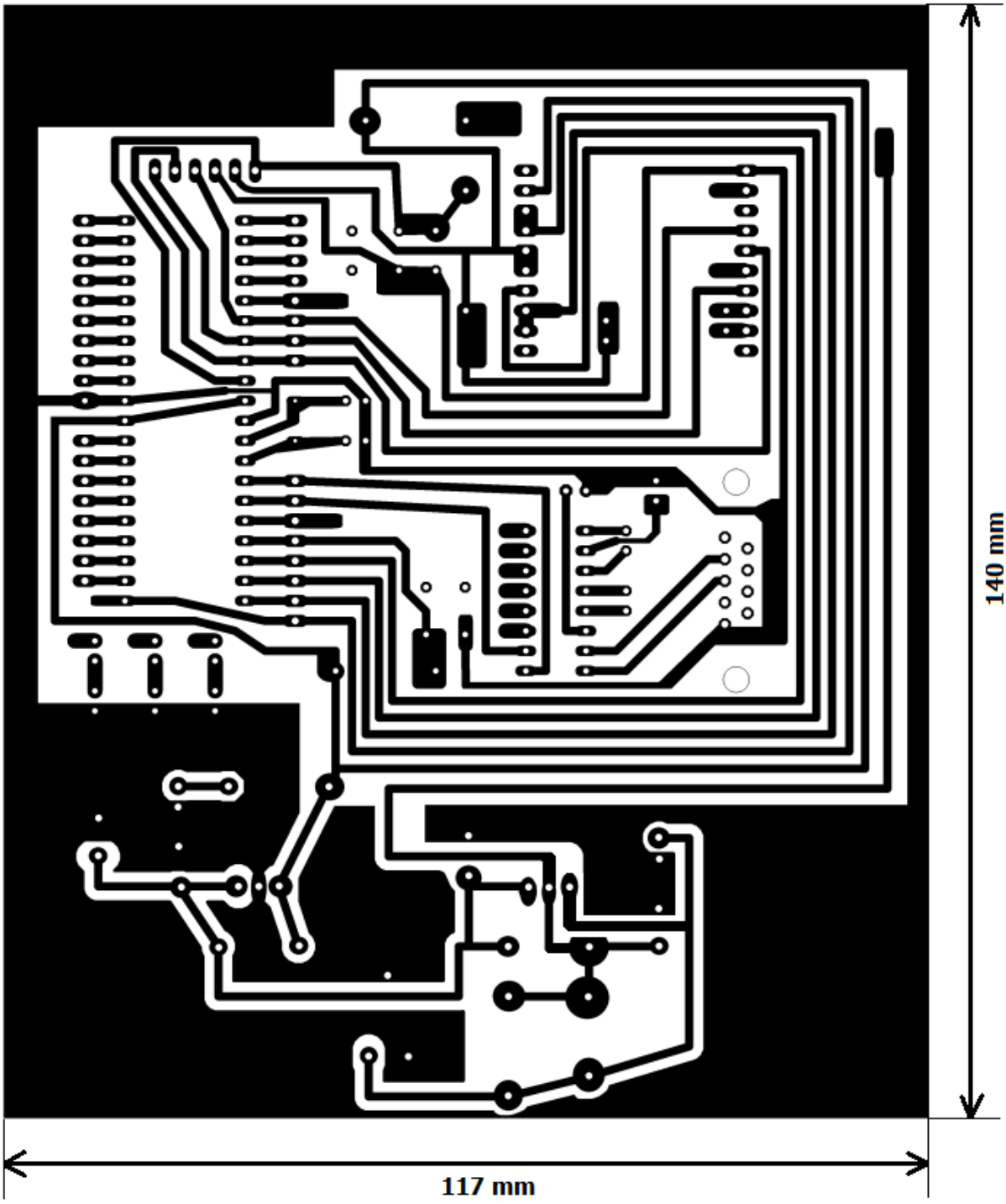
Obrázek B.2: *Layout DPS adaptéru RFM12BP*

Příloha C: *Schéma zapojení radiostanice*

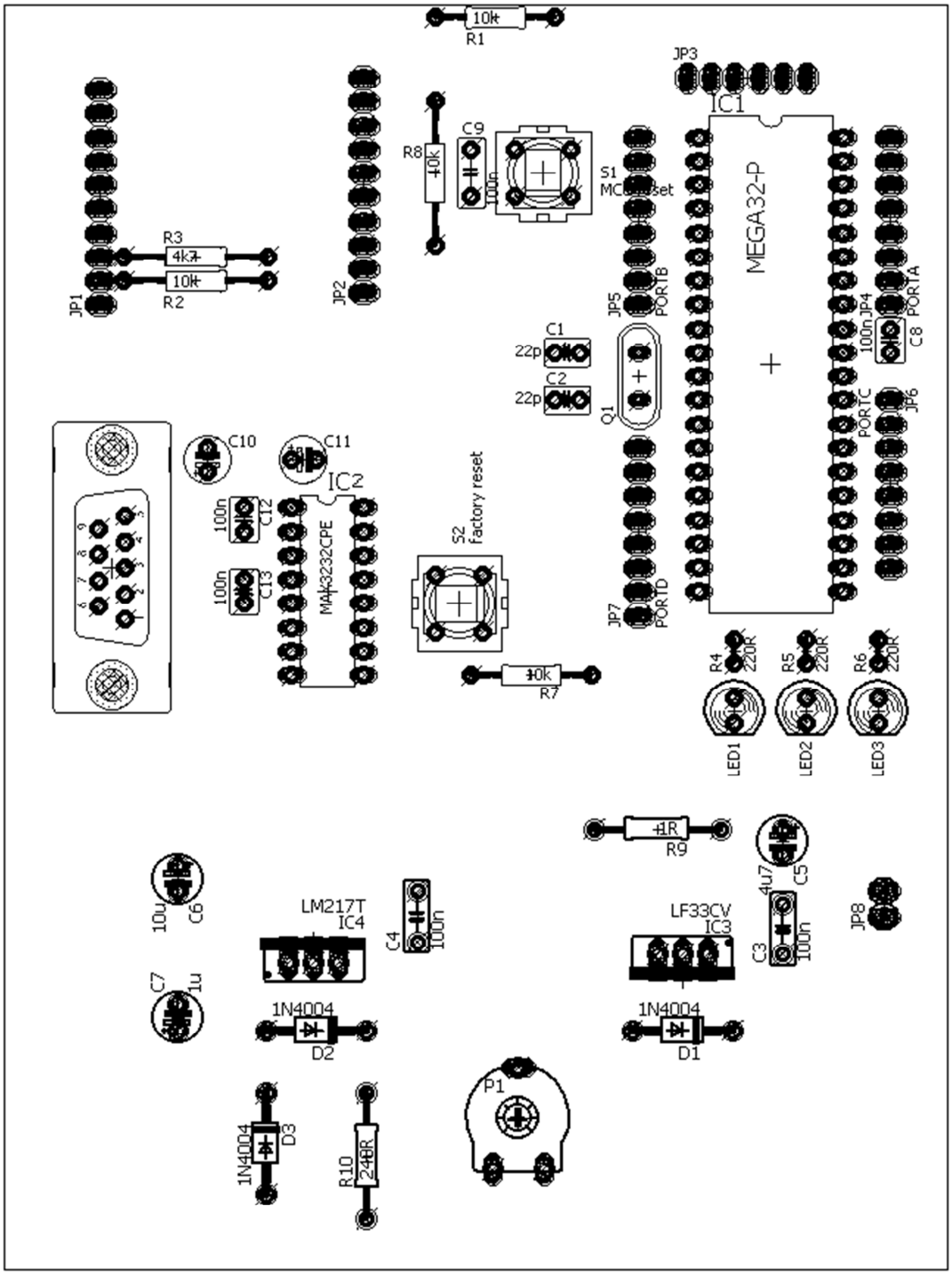


Obrázek C.1: *Celkové schéma zapojení radiostanice*

Příloha D: *Layout DPS radiostanice a osazovací plán*



Obrázek D.1: *Layout DPS*



Obrázek D.1: *Osazovací plán*

Příloha E: *Seznam součástek*

Tabulka E.1: *Seznam použitých součástek*

označení	typ	hodnota	Poznámka
rezistory, trimry			
R1	0207	10k	pull-up nRES
R2	0207	10k	pull-up FSK
R3	0207	4k7	pull-up nIRQ
R4	0207	220R	LED1 (žlutá)
R5	0207	220R	LED2 (zelená)
R6	0207	220R	LED3 (červená)
R7	0207	10k	pull-up tl. FACTORY RESET
R8	0207	10k	pull-up tl. MCU RESET
R9	0207	1R	výstup LF33CV
R10	0207	240R	výstup LM217T
P1	PT10VK005	5k	nastavení LM217T
kondenzátory			
C1	keramický	22p	XTAL MCU
C2	keramický	22p	
C3	fóliový	100n	vstup LF33CV
C4	fóliový	100n	vstup LM217T
C5	elektrolytický	4u7	výstup LF33CV
C6	elektrolytický	10u	zapojení LM217T
C7	elektrolytický	1u	výstup LM217T
C8	keramický	100n	Aref MCU
C9	fóliový	100n	tl. RESET MCU
C10	elektrolytický	0,1u	zapojení MAX3232CPE
C11	elektrolytický	0,1u	
C12	keramický	100n	
C13	keramický	100n	
polovodiče a IO			
D1	1N4004	-	zapojení LF33CV
D2	1N4004	-	zapojení LM217T
D3	1N4004	-	
LED1		žlutá	USART aktivita
LED2		zelená	Rx aktivní
LED3		červená	Tx aktivní
IC1	Atmel AVR Mega 32A	-	MCU (PDIP40)
IC2	MAX3232CPE	-	převodník (PDIP16)
IC3	LF33CV	3,3V/ 1 A	stabilizátor 3,3V
IC4	LM217T	1,2 - 37 V / 2,2 A	
Q1	HC49/U	9,216 MHz	krystal MCU

Tabulka pokračuje na další straně.

Tabulka E.1: *Seznam použitých součástek (pokračování)*

označení	typ	hodnota	Poznámka
S1	TC-0103-T	-	MCU RESET
S2	TC-0103-T	-	FACTORY RESET
X1	CAN9Z90	-	RS232 konektor
JP1, JP2	SOKL 20	-	patice adaptéru RFM12BP
JP3	oboustranný kolík	6 x 2,54 mm	ISP patice
JP4	oboustranný kolík	8 x 2,54 mm	PORT A
JP5	oboustranný kolík	8 x 2,54 mm	PORT B
JP6	oboustranný kolík	8 x 2,54 mm	PORT C
JP7	oboustranný kolík	8 x 2,54 mm	PORT D
JP8	AKZ692/2-2.54-V	-	napájecí svorkovnice

Příloha F: *Průběhy napětí převodníků RS232*

Integrovaný RS232
Základní deska MSI P43T-C51

PCI karta
Axago PCEA-PS

Express Card
I-Tec

USB/RS232 převodník
HL-340

Obrázek F.1: *Průběhy napětí linek jednotlivých RS232 převodníků*

Příloha G: *Zdrojové kódy v programovacím jazyku C*

Funkce pro SPI komunikaci je upravenou verzi publikovanou v datasheetu[13] na straně 141 pro přenos 8 nebo 16 bitových čísel (nastavuje se na základě parametru "**IS_16BIT_NUM**")

```
unsigned char SPI_Data(unsigned char DATA_TO_SPI, unsigned char
IS_16BIT_NUM){
unsigned char DATA_FROM_SPI = 0;
SS = 0;                          //SS=0 (zahajeni komunikace)
SPDR = DATA_TO_SPI;   //start transmission from Master to Slave
while (!(SPSR & (1<<SPIF)));     //Wait fortransmission complete
if (IS_16BIT_NUM == 0){          //Pokud nebude 16bit cislo
    SS = 1;                      //SS=1 (komunikace neaktivni)
}
DATA_FROM_SPI = SPDR;
delay_us(1);
return DATA_FROM_SPI;
}
```

Funkce pro zapnutí přijímače RFM12BP, včetně povolení přístupu k FIFO registru:

```
void RFM12_Receiver_On(){
RX_LED = 1;                      //Zapnuti LED a HW prijimace
RFM12_RXEN = 1;
SPI_Data(0x82,1);                //RFM12 Power Management Cmd.
SPI_Data(0xC9,0);                //Zapnuti prijimace
SPI_Data(0xCA,1);                //RFM12 FIFO and Reset Mode Cmd.
SPI_Data(0x83,0);                //Povoleni FIFO registru
//ceka dokud je nIRQ=0 => dokonceni vnitrnich operaci RFM12
while (nIRQ == 0){}
return;
}
```

Funkce pro vypnutí přijímače RFM12BP včetně zakázání přístupu k FIFO registru:

```
void RFM12_Receiver_Off(){
RX_LED = 0;                         //Vypnuti LED a HW prijimace
RFM12_RXEN = 0;
SPI_Data(0xCA,1);                   //RFM12 FIFO and Reset Mode Cmd.
SPI_Data(0x81,0);                   //Zakazani FIFO registru
SPI_Data(0x82,1);                   //RFM12 Power Management Cmd.
SPI_Data(0x08,0);                   //Vypnuti vysilace a prijimace
return;
}
```

Funkce pro načítání STATUS registru rádiového modulu:

```
void RFM12_ReadStatus(){
RFM_STATUS_HIGH = SPI_Data(0x00,1);
RFM_STATUS_LOW = SPI_Data(0x00,0);
return;
}
```

Funkce pro Power on Reset test rádiového modulu:

```
void RFM12_Por_Status(){
if (nIRQ == 0){
     RFM12_ReadStatus();              //nacti STATUS registr
     if (RFM_STATUS_HIGH == 0x40){
          putsf("RFM12 POR [OK]");    //RFM12 v poradku
     }else{
          putsf("RFM12 POR [fail]"); //RFM12spatny
     }
}else{
     putsf("RFM12 nIRQ line high");   //RFM12 chyba nIRQ
}
return;
}
```

Funkce pro inicializaci radiového modulu podle popisu v kapitole 2.4.2. Parametry instrukci jsou popsány podrobně v příloze A.

```
void RFM12_Init(){
delay_ms(850);                  //cekej dokudneni hotovy POR
//pevne nastavene parametry - nemenitelne za behu programu
SPI_Data(0x00,1);               //nulovy byte - zacatek konfigurace
SPI_Data(0x00,0);
SPI_Data(0x80,1);               //RFM12 Configuration Setting Cmd.
SPI_Data(0xD8,0);
SPI_Data(0xC2,1);               //RFM12 Data Filter Cmd. 0xC2
SPI_Data(0xAC,0);
SPI_Data(0xCA,1);               //RFM12 FIFO and Reset Mode Cmd.
SPI_Data(0x81,0);
SPI_Data(0xE0,1);               //RFM12 Wake Up Timmer Cmd.
SPI_Data(0x00,0);
SPI_Data(0xC8,1);               //RFM12 Low Duty Cycle Cmd.
SPI_Data(0x00,0);
SPI_Data(0xC4,1);               //RFM12 AFC Cmd.
SPI_Data(0xF7,0);
//uzivatelsky nastavene parametry z EEPROM ctene pres SRAM registry
SPI_Data(0xC6,1);               //RFM12 Data Rate Cmd.
SPI_Data(RFM_DATARATE,0);       //SRAM registr
SPI_Data(RFM_FREQ_1,1);         //RFM12 Frequency Setting Cmd.
SPI_Data(RFM_FREQ_2,0);         //SRAM registr
SPI_Data(0xCE,1);               //RFM12 Synchron Pattern Cmd.
SPI_Data(RFM_SYNCPAT,0);        //SRAM registr
SPI_Data(DEFAULT_RFM_TX1,1);    //RFM12 Tx Conf. Control Cmd.
SPI_Data(RFM_TX,0);             //SRAM registr
SPI_Data(DEFAULT_RFM_RX1,1);    //RFM12 Receiver Control Cmd.
SPI_Data(RFM_RX,0);             //SRAM registr
return;
}
```

Funkce pro odesílání dat (rádiového paketu) s podporou přerušení nIRQ, které zajišťuje správné časování s ohledem na zvolenou přenosovou rychlost:

```
void RFM12_Send_Data(unsigned char *DATA_TO_SEND){
unsigned char i = 0;
RFM12_TXEN = 1;                  //Zapnuti LED a HW vysilace
TX_LED = 1;
SPI_Data(0x82,1);                //RFM12 Power Management Cmd.
SPI_Data(0x39,0);                //Zapnuti vysilace a prijimace
//ceka dokud neni nIRQ=0 => zada o preruseni (dalsi Tx data)
while (nIRQ == 1){}

//Synchronizace
SPI_Data(0xB8,1);                //RFM12 zapis do Tx registru
SPI_Data(0xAA,0);
while (nIRQ == 1){}
SPI_Data(0xB8,1);                //RFM12 zapis do Tx registru
SPI_Data(0xAA,0);
while (nIRQ == 1){}

//Identifikace
SPI_Data(0xB8,1);                //RFM12 zapis do Tx registru
SPI_Data(0x2D,0);                //1st
while (nIRQ == 1){}
SPI_Data(0xB8,1);                //RFM12 zapis do Tx registru
SPI_Data(RFM_SYNCPAT,0);         //2nd (default: 0x4B)
while (nIRQ == 1){}
SPI_Data(0xB8,1);                //RFM12 zapis do Tx registru
SPI_Data(0xCA,0);                //vlastni identifikace (0xCA)
while (nIRQ == 1){}
```

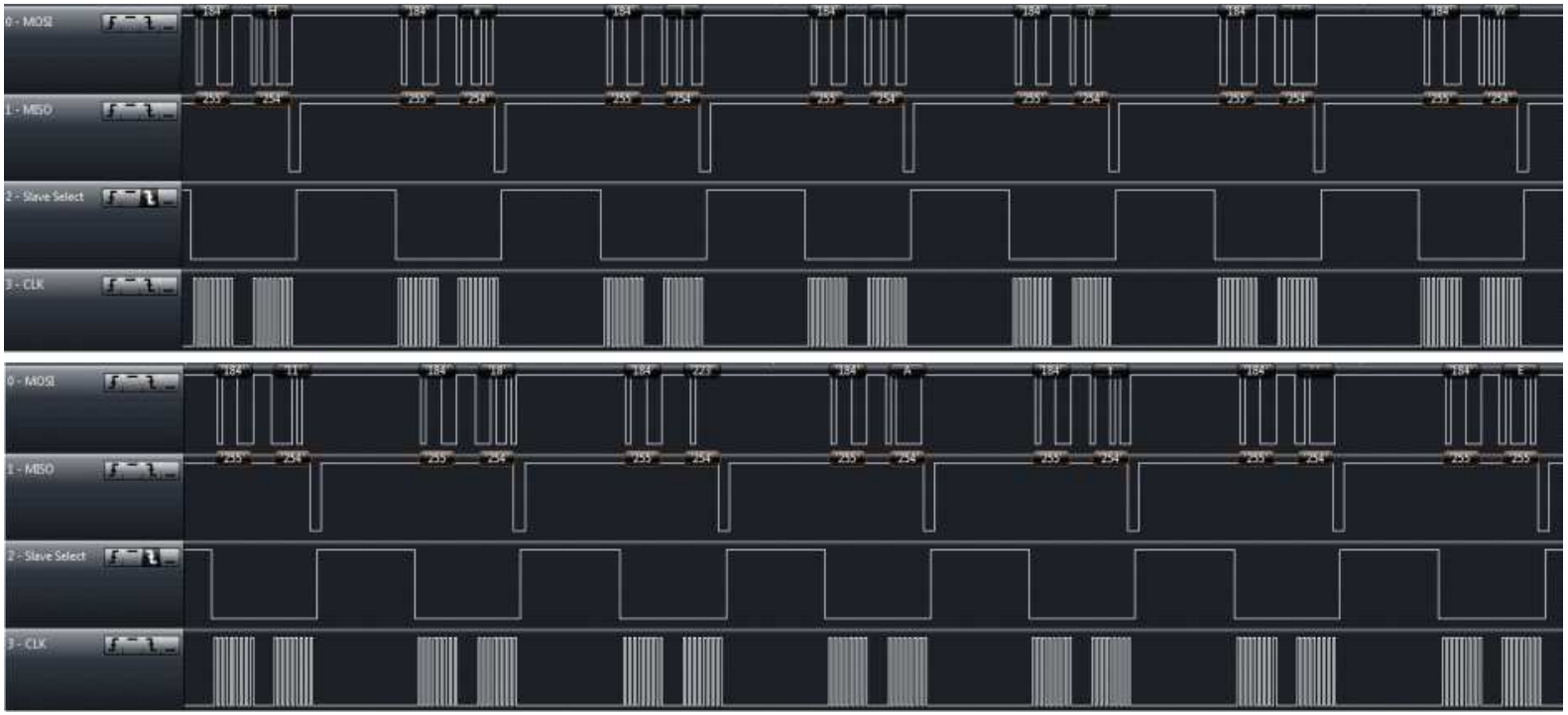
```
//DATA K PRENOSU
for (i=0;i<19;i++){
      SPI_Data(0xB8,1);                //RFM12 zapis do Tx registru
      SPI_Data(DATA_TO_SEND[i],0);     //DATA K PRENOSU
      while (nIRQ == 1){}
}

//Ukonceni paketu
SPI_Data(0xB8,1);                //RFM12 zapis do Tx registru
SPI_Data(0xAA,0);                //1st
while (nIRQ == 1){}
SPI_Data(0xB8,1);                //RFM12 zapis do Tx registru
SPI_Data(0xAA,0);                //2nd
while (nIRQ == 1){}

SPI_Data(0x82,1);                //RFM12 Power Management Cmd.
SPI_Data(0x08,0);                //Vypnuti vysilace a prijimace
RFM12_TXEN = 0;                  //Vypnuti LED a HW vysilace
TX_LED = 0;
return;
}
```

Funkce pro příjem dat (rádiového paketu) s podporou přerušení nIRQ, které zajišťuje správné časování s ohledem na zvolenou přenosovou rychlost:

```
void RFM12_Data_Received (unsigned char *INPUT_DATA){
unsigned char i = 0;
for (i=0;i<20;i++){
      SPI_Data(0xB0,1);                    //RFM12 cteni Rx FIFO
      INPUT_DATA[i] = SPI_Data(0x00,0);    //SPI clk
      while (nIRQ == 1){}                  //Cekej na dalsi nIRQ
}
RFM12_Receiver_Off();                      //Vypnuti prijimace
RFM12_Receiver_On();                       //Zapnuti prijimace
return;
}
```

Příloha H: *Ukázka komunikace na sběrnici SPI*

V prvním případě se jedná o ukázku nešifrované komunikace, kde vidíme prvních 7 znaků zprávy "*Hello Word! 123*". V druhém případě je komunikace již zašifrována.



Obrázek H.1: *Komunikace zachycená na sběrnici SPI*